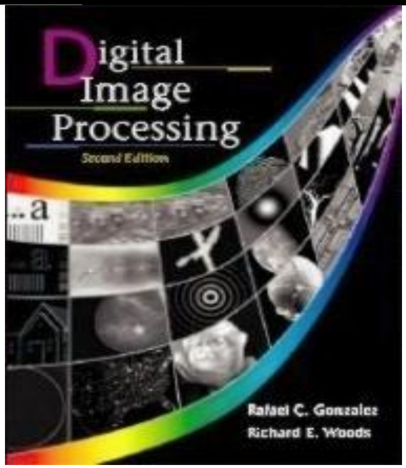
Digital Image Processing: Introduction

References



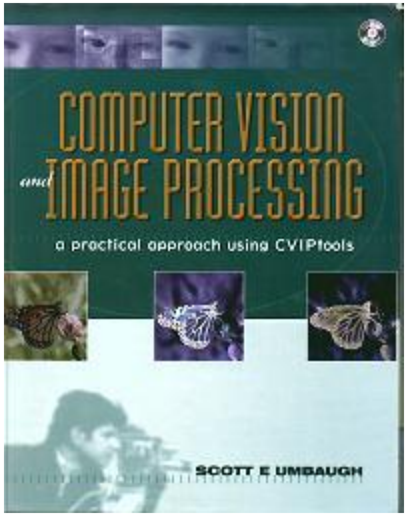
“Digital Image Processing”, Rafael C. Gonzalez & Richard E. Woods, Addison-Wesley, 2002

or

“Computer Vision and image processing A practical approach using cvip tools

Scott E umbaugh

Prentice hall 1998



Contents

This lecture will cover:

- What is a digital image?
- What is digital image processing?
- History of digital image processing
- State of the art examples of digital image processing
- Key stages in digital image processing

Computer imaging

- It’s defined as the acquisition and processing of visual information by computer.
- The ultimate receiver of information is:
  - Computer
  - Human visual system
- So we have two categories:-
  - Computer vision
  - Image processing

Computer vision and image processing

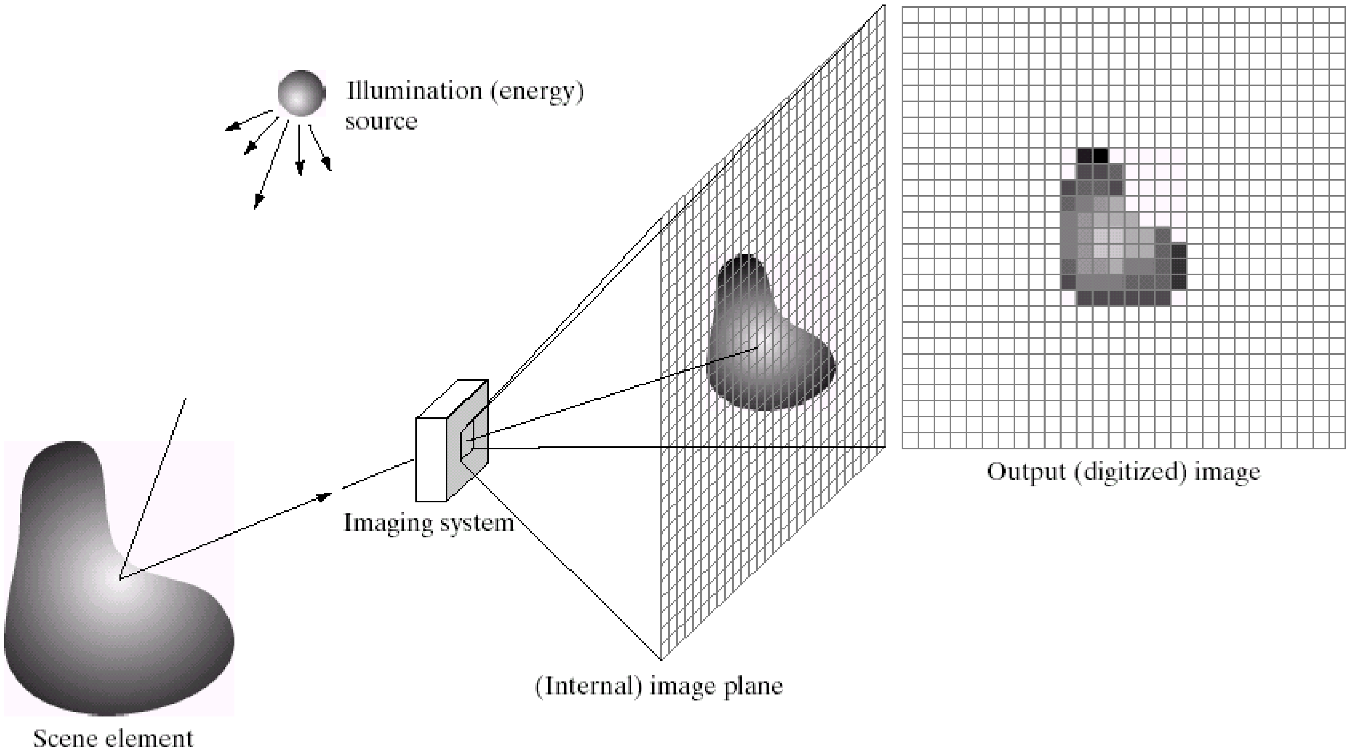
- In computer vision:
  - The processed output images are for use by computer.
- In Image processing:
  - The output images are for human consumption

Computer vision

- One of the computer vision fields is image analysis.
- It involves the examination of image data to facilitate solving a vision problem.
- Image analysis has 2 topics:
  - Feature extraction: acquiring higher level image information
  - Pattern classification taking these higher level of information and identifying objects within the image

What is a Digital Image?

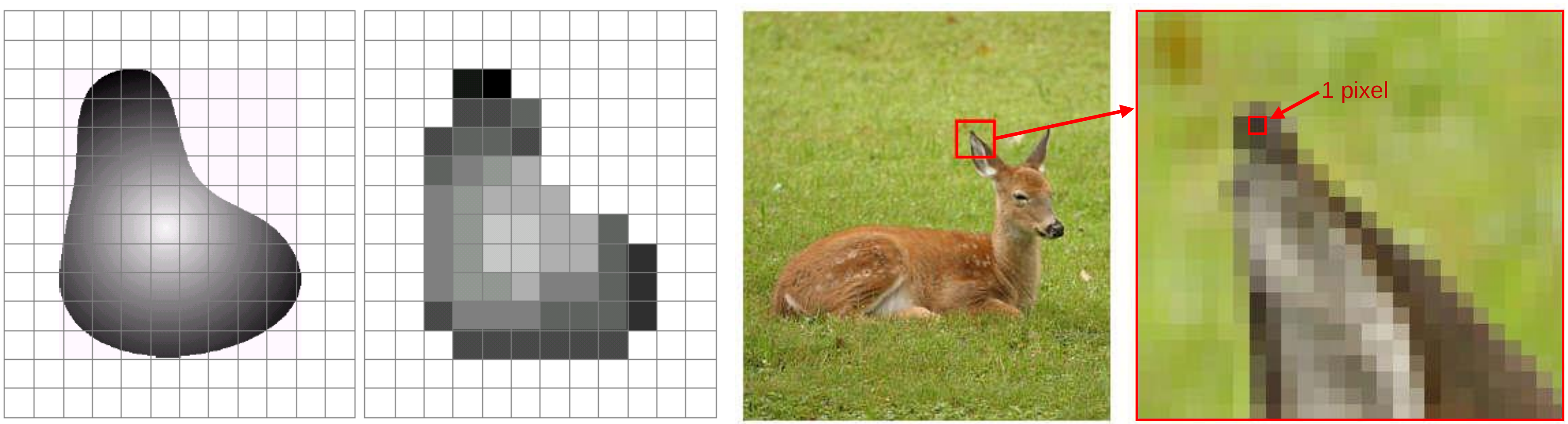
A **digital image** is a representation of a two-dimensional image as a finite set of digital values, called picture elements or pixels



What is a Digital Image? (cont…)

Pixel values typically represent gray levels, colours, heights, opacities etc

Remember *digitization* implies that a digital image is an *approximation* of a real scene



What is a Digital Image? (cont…)

Common image formats include:

- 1 sample per point (B&W or Grayscale)
- 3 samples per point (Red, Green, and Blue)

For most of this course we will focus on grey-scale images

What is Digital Image Processing?

Digital image processing focuses on two major tasks

- Improvement of pictorial information for human interpretation
- Processing of image data for storage, transmission and representation for autonomous machine perception(ادراك)

Some argument about where image processing ends and fields such as image analysis and computer vision start

What is DIP? (cont…)

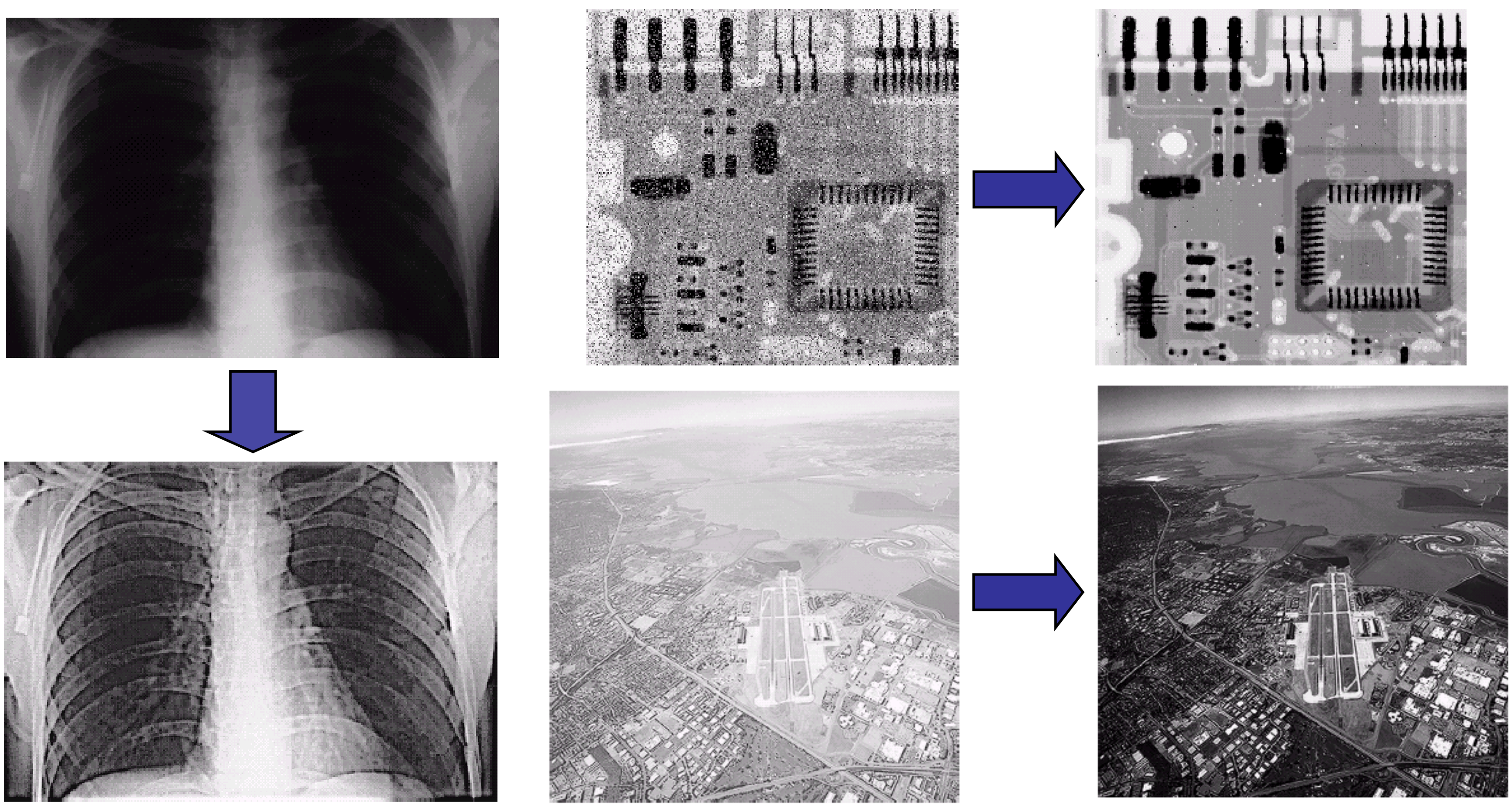
The continuum (متسلسل - متصل)from image processing to computer vision can be broken up into low-, mid- and high-level processes

Low Level Process	Mid Level Process	High Level Process
Input: Image **Output:** Image **Examples:** Noise removal, image sharpening	**Input:** Image **Output:** Attributes **Examples:** Object recognition, segmentation	**Input:** Attributes **Output:** Understanding **Examples:** Scene understanding, autonomous navigation

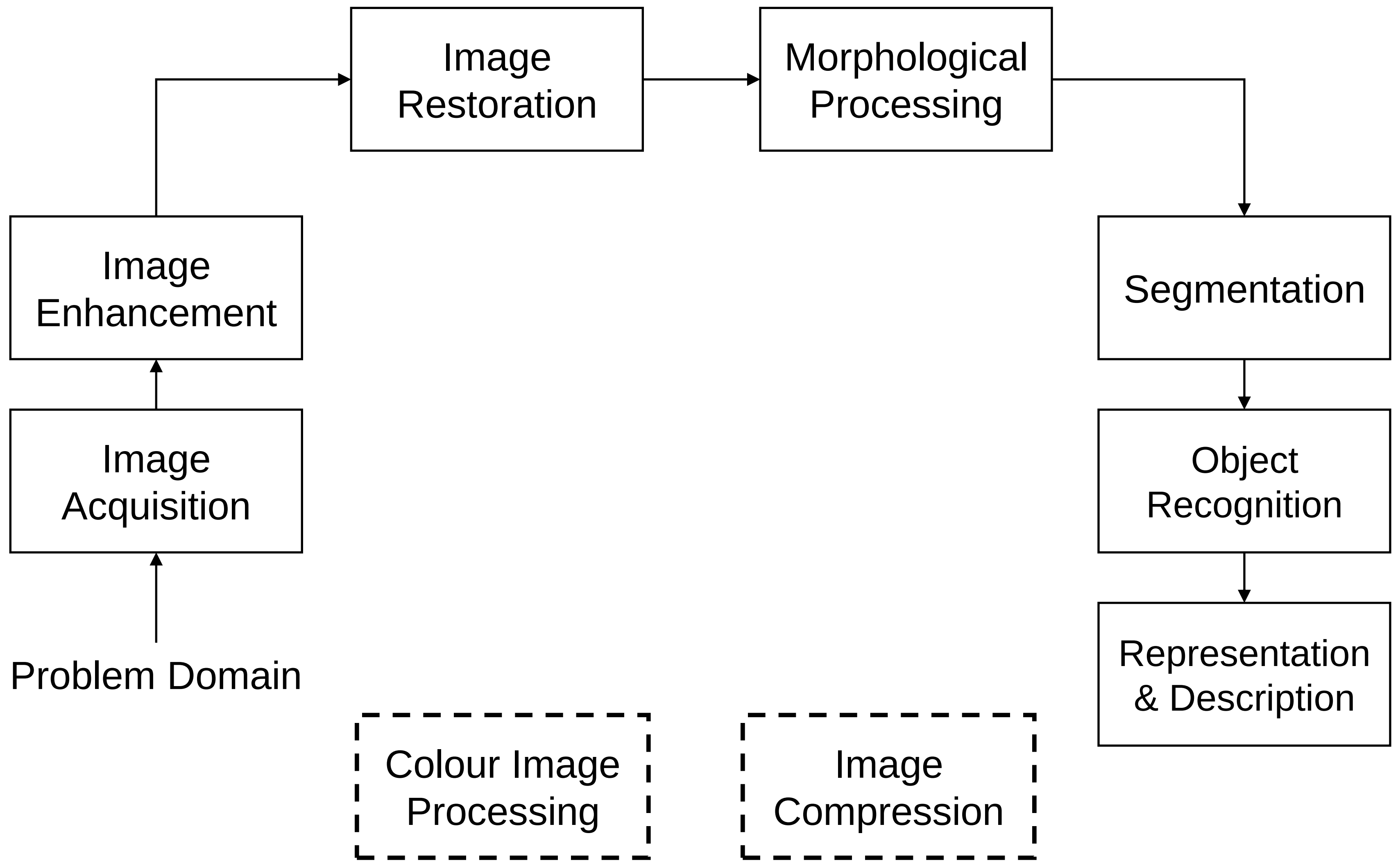
In this course we will
stop here

Examples: Image Enhancement

One of the most common uses of DIP techniques: improve quality, remove noise etc



Key Stages in Digital Image Processing



Key Stages in Digital Image Processing: Image Aquisition

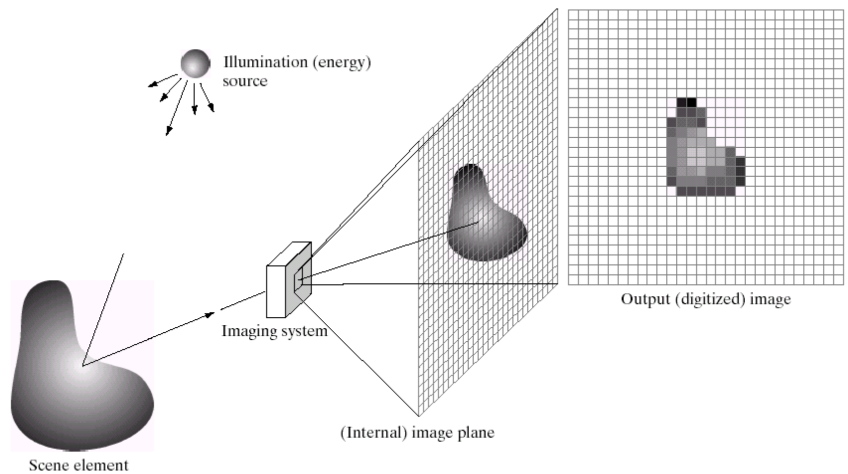
Image Restoration

Morphological Processing

Image Enhancement

Segmentation

Image Acquisition



Object Recognition

Representation & Description

Problem Domain

Colour Image Processing

Image Compression

Image Enhancement: taking an image and improving it visually

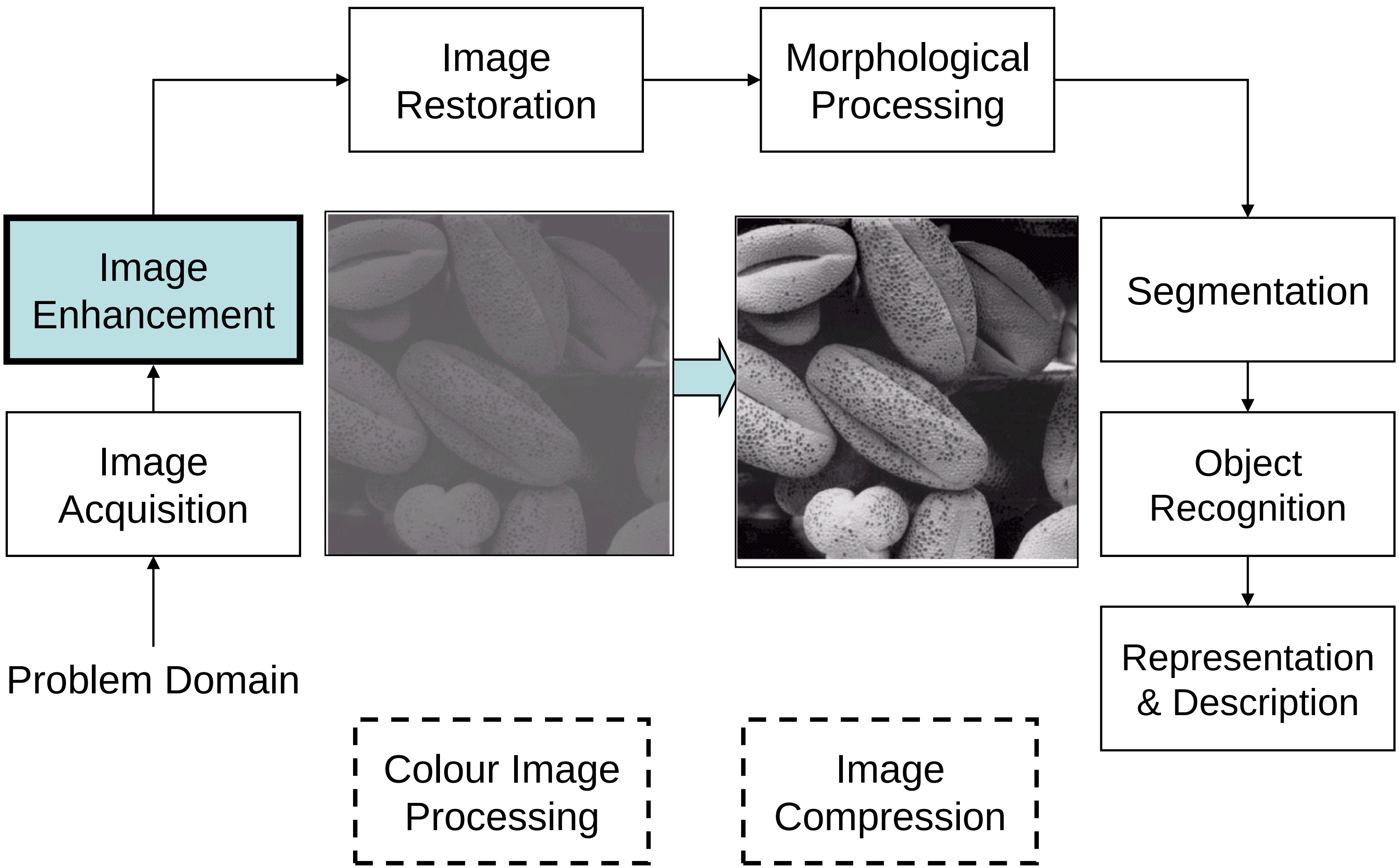
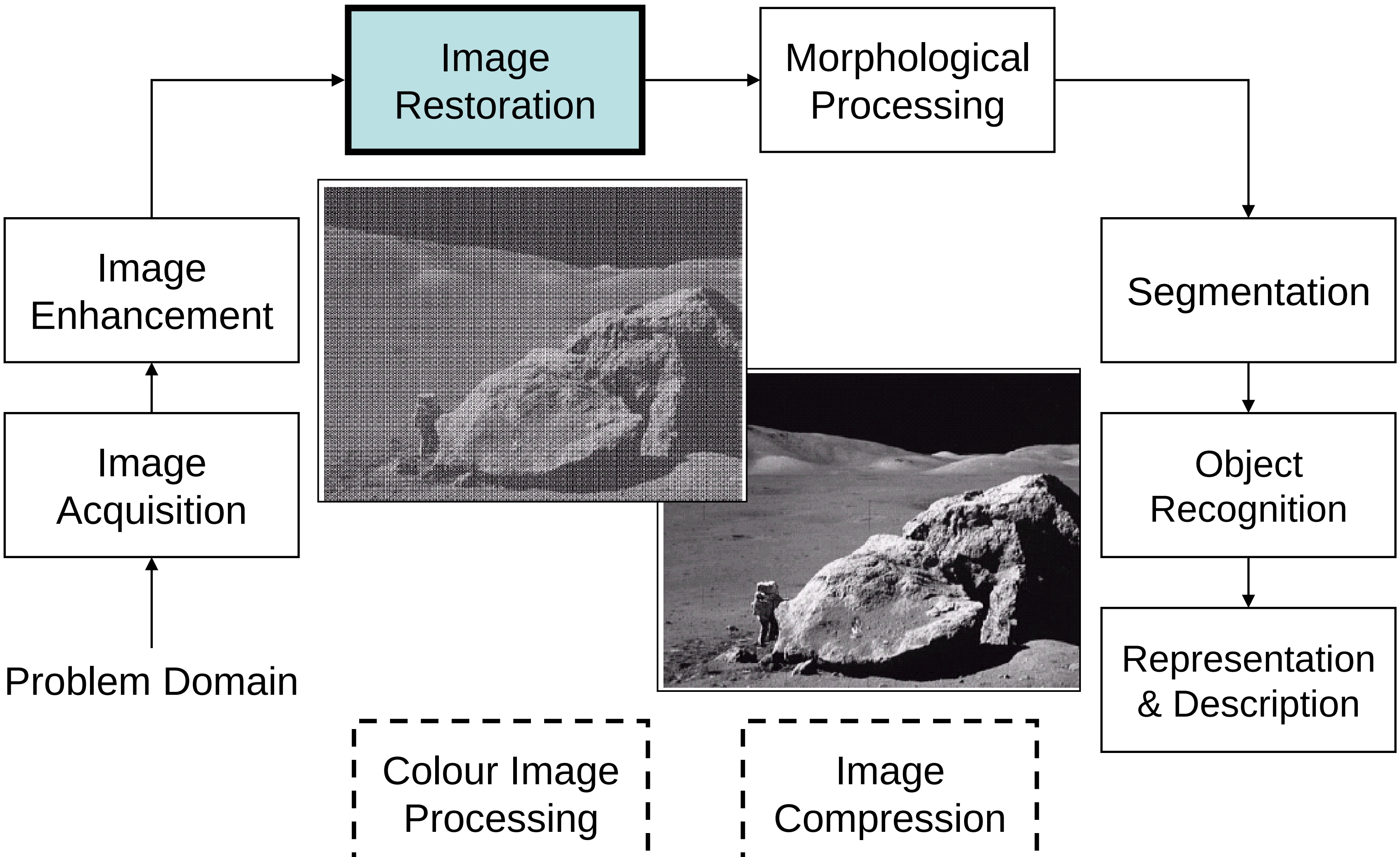
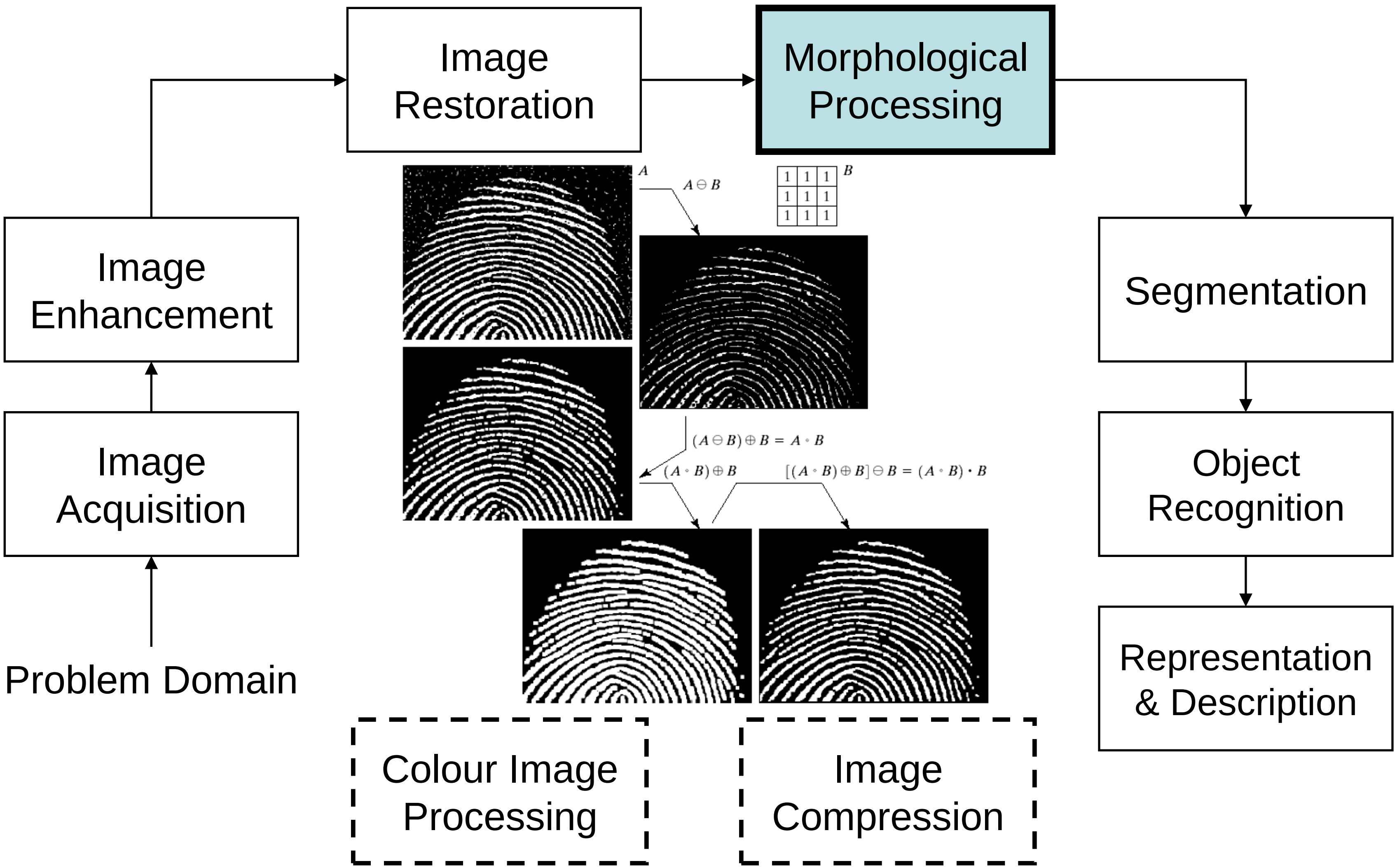


Image Restoration :

taking an image with some known or estimated degradation and restoring it to its original appearing



Key Stages in Digital Image Processing: Morphological Processing



Key Stages in Digital Image Processing: Segmentation

Image Acquisition

Image Enhancement

Image Restoration

Morphological Processing

Segmentation

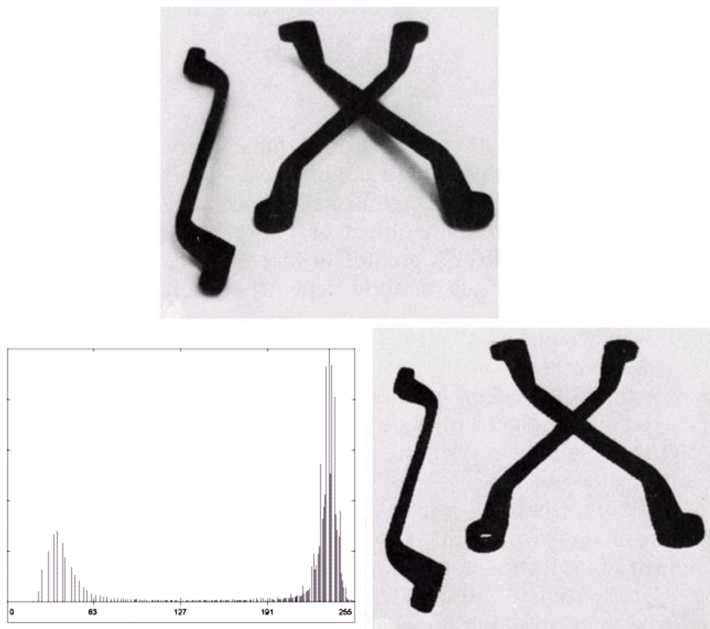
Object Recognition

Representation & Description

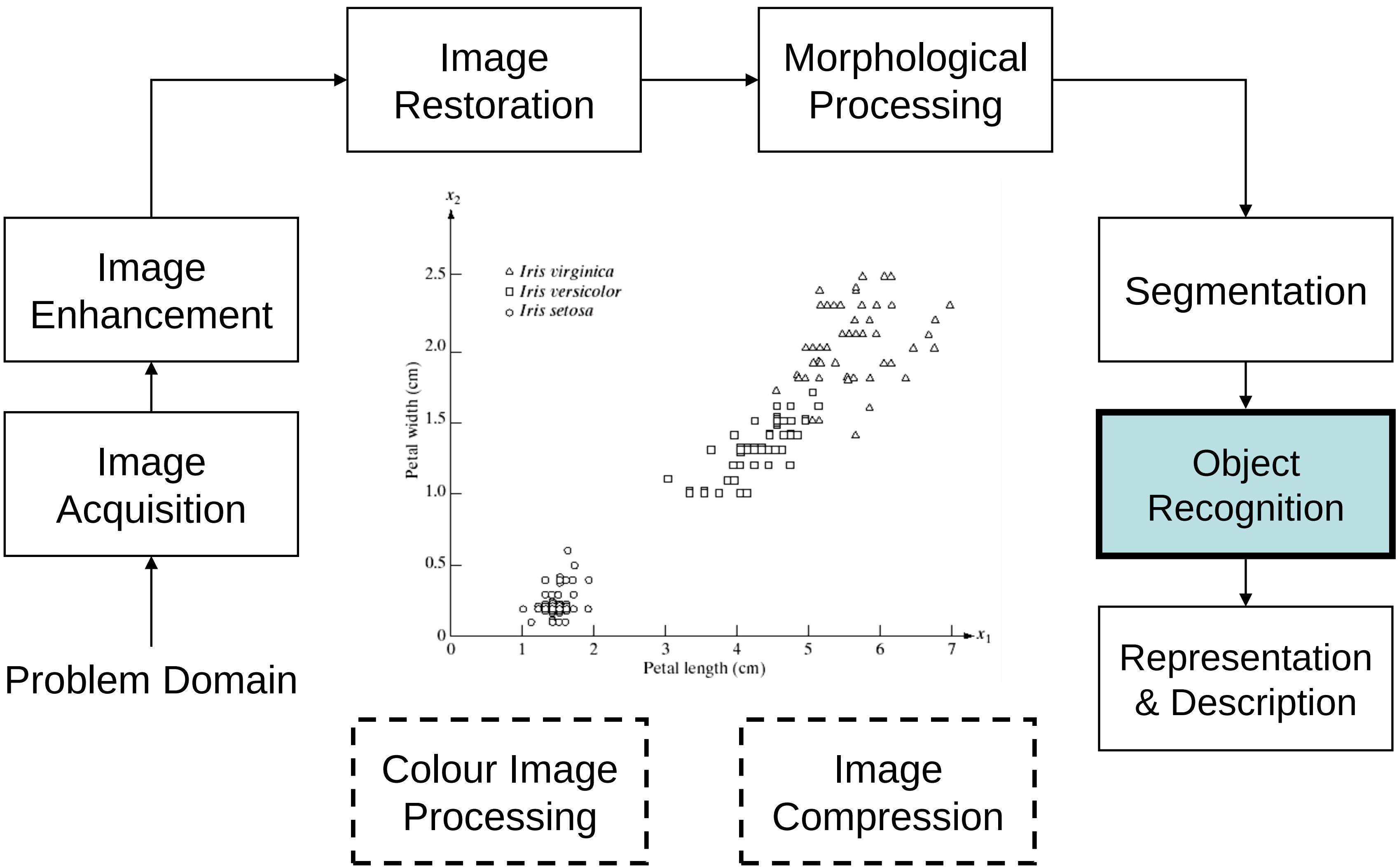
Problem Domain

Colour Image Processing

Image Compression



Key Stages in Digital Image Processing: Object Recognition



Key Stages in Digital Image Processing: Representation & Description

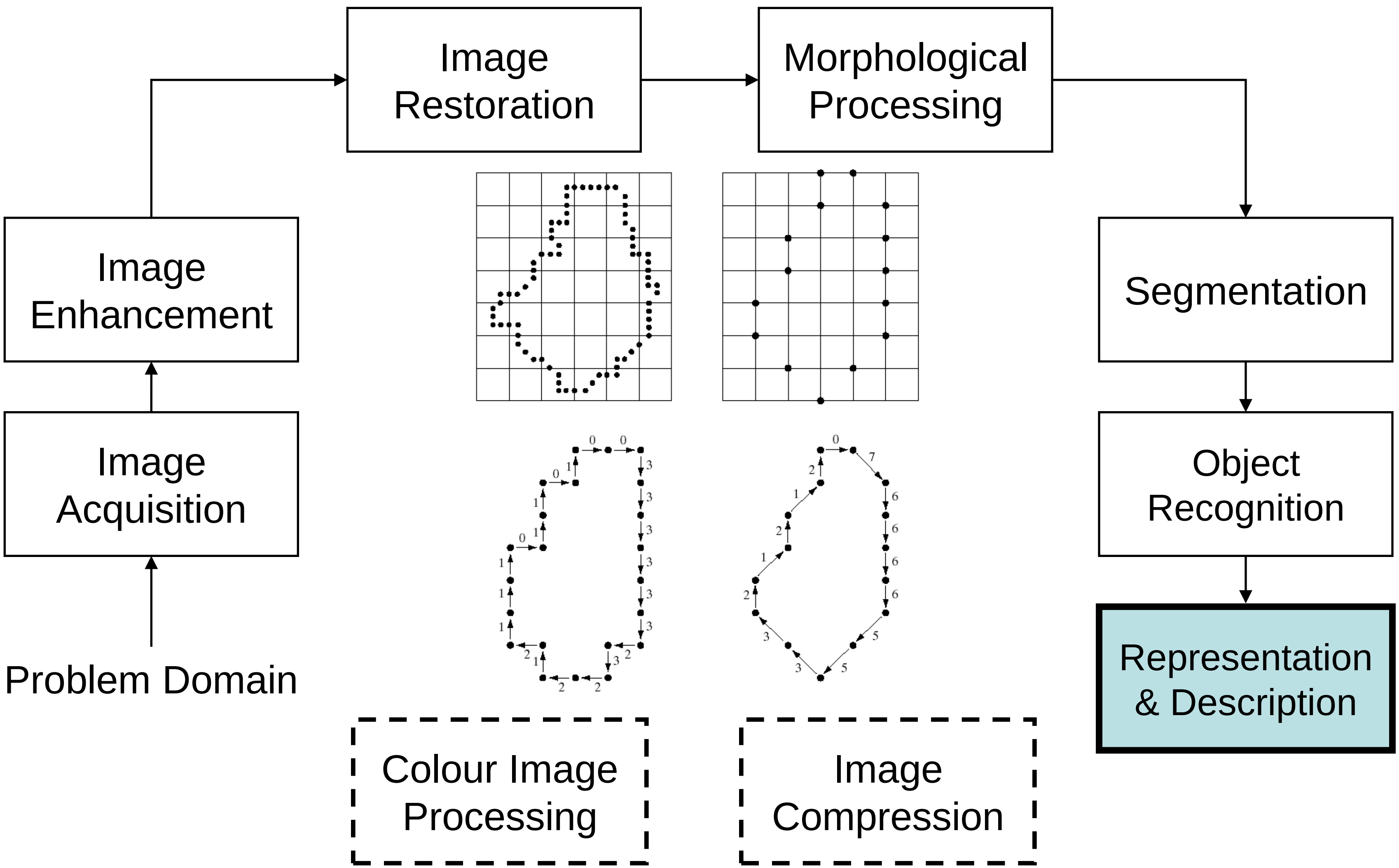
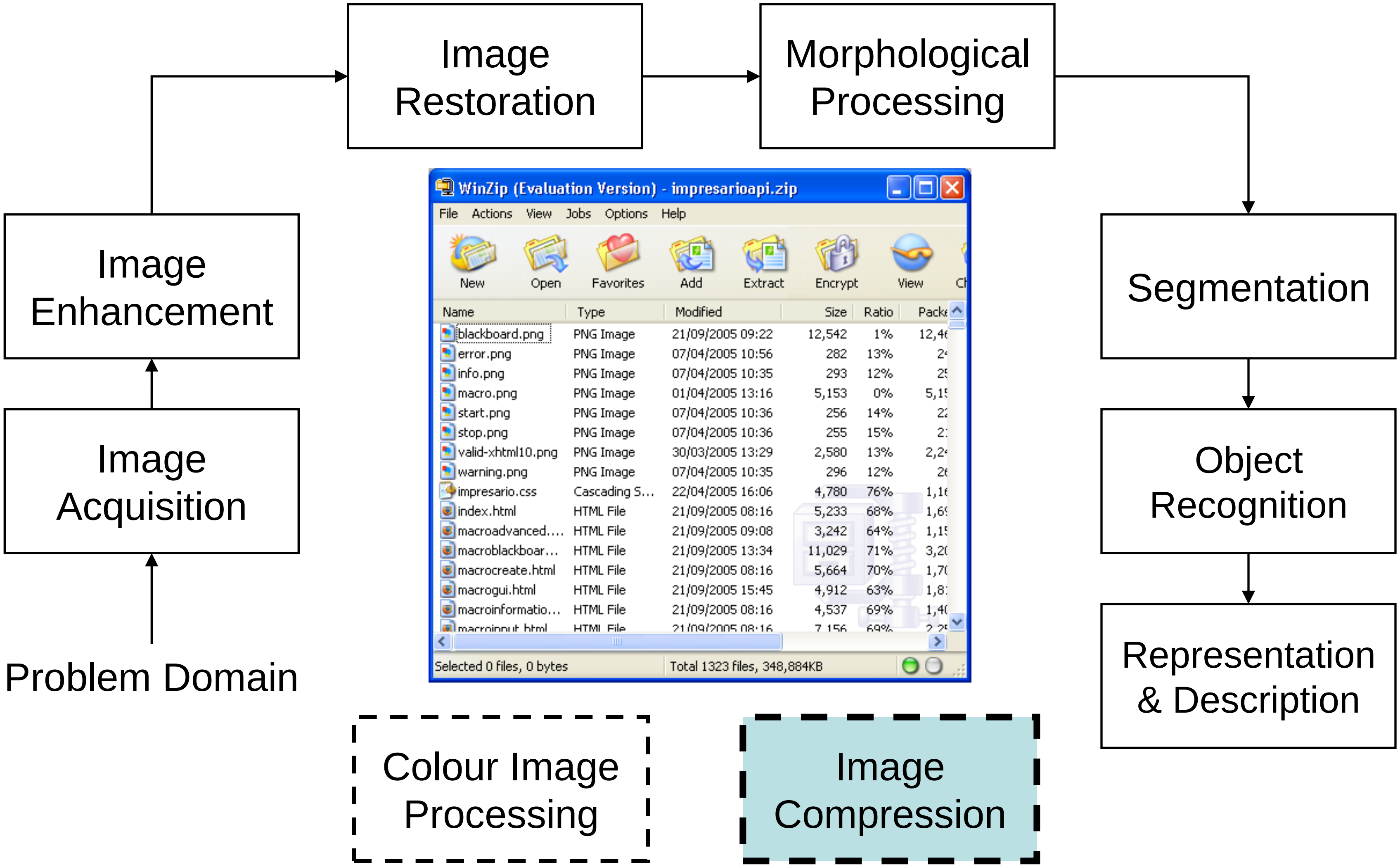
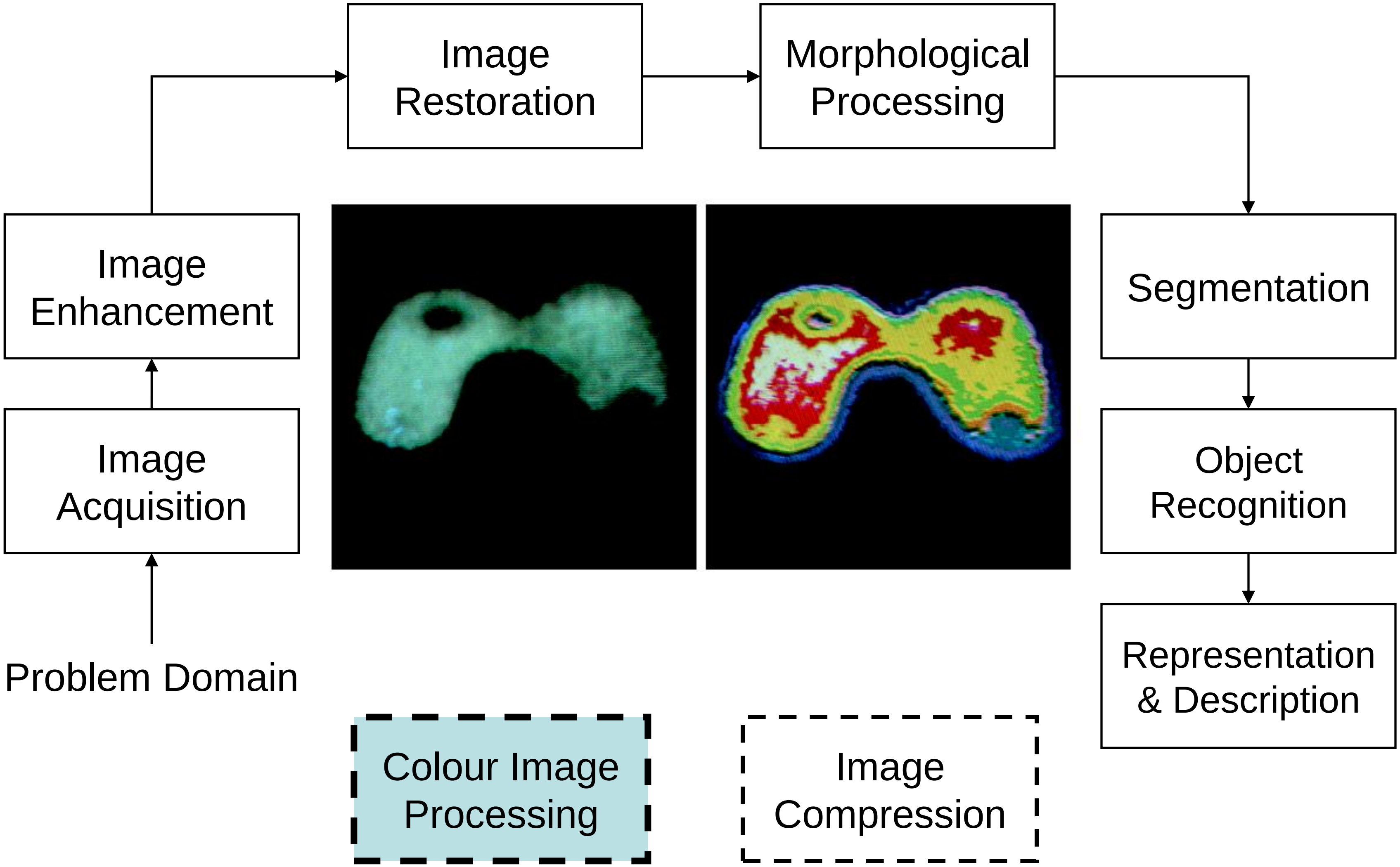


Image compression:

reducing the massive amount of data needed to represent an image



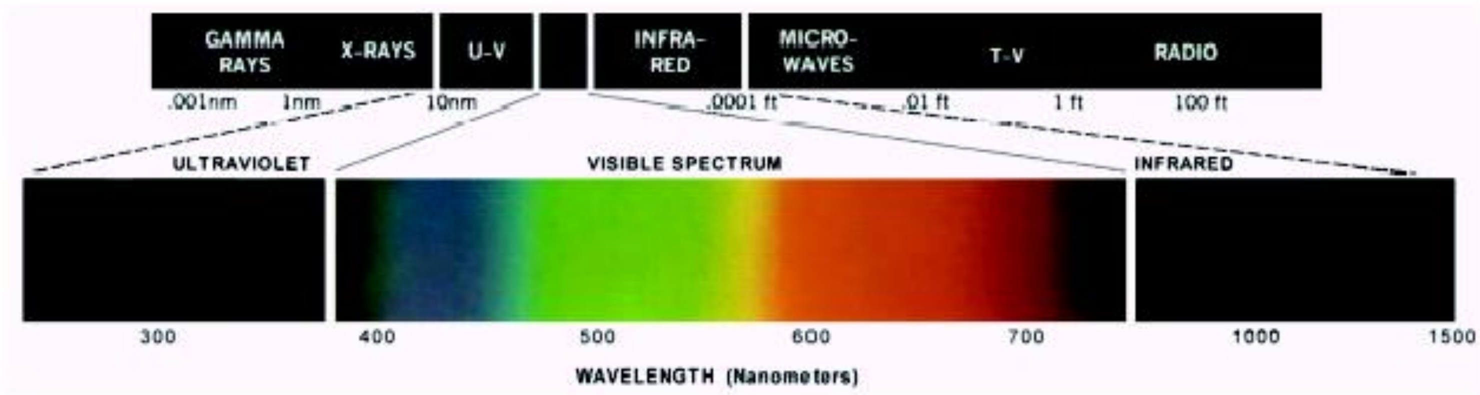
Key Stages in Digital Image Processing: Colour Image Processing



Light and the Electromagnetic Spectrum

Light is just a particular part of the electromagnetic spectrum that can be sensed by the human eye

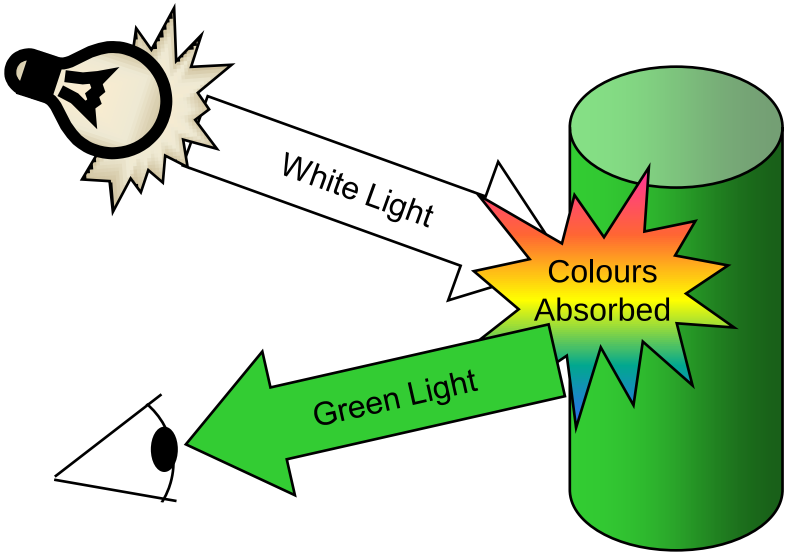
The electromagnetic spectrum is split up according to the wavelengths of different forms of energy



Reflected Light

The colours that we perceive are determined by the nature of the light reflected from an object

For example, if white light is shone onto a green object most wavelengths are absorbed, while green light is reflected from the object



Sampling, Quantisation and Resolution

In the following slides we will consider what is involved in capturing a digital image of a real-world scene

– Image sensing and representation
– Sampling and quantisation
– Resolution

Image Representation

- Before we discuss image acquisition recall that a digital image is composed of M rows and N columns of pixels each storing a value
- Pixel values are most often grey levels in the range 0-255(black-white)
- We will see later on that images can easily be represented as matrices

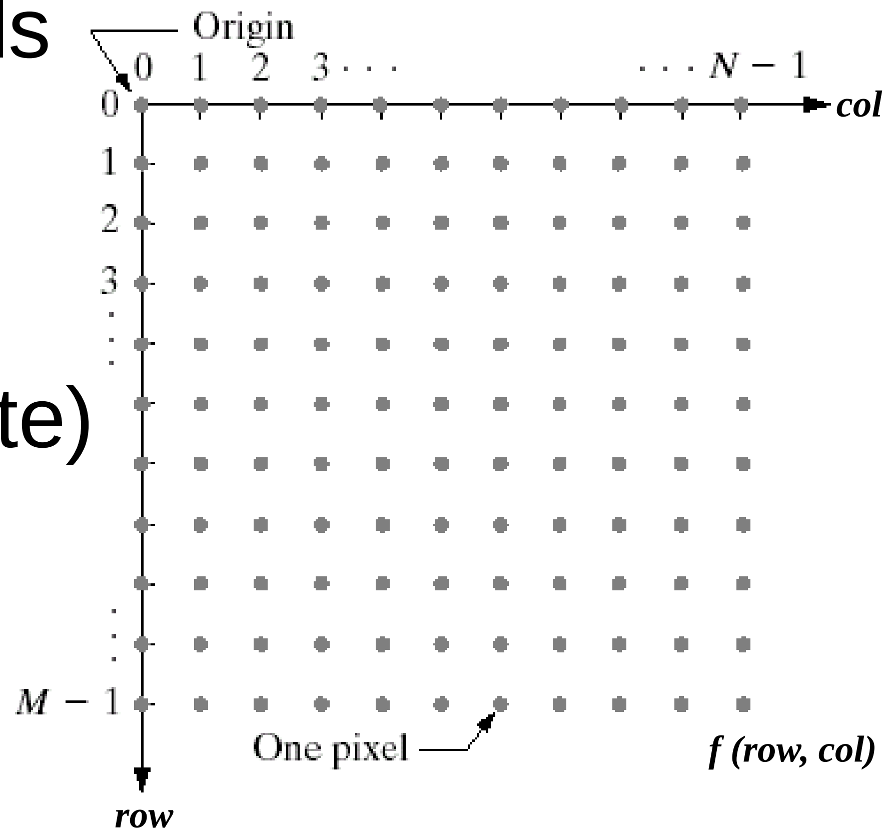
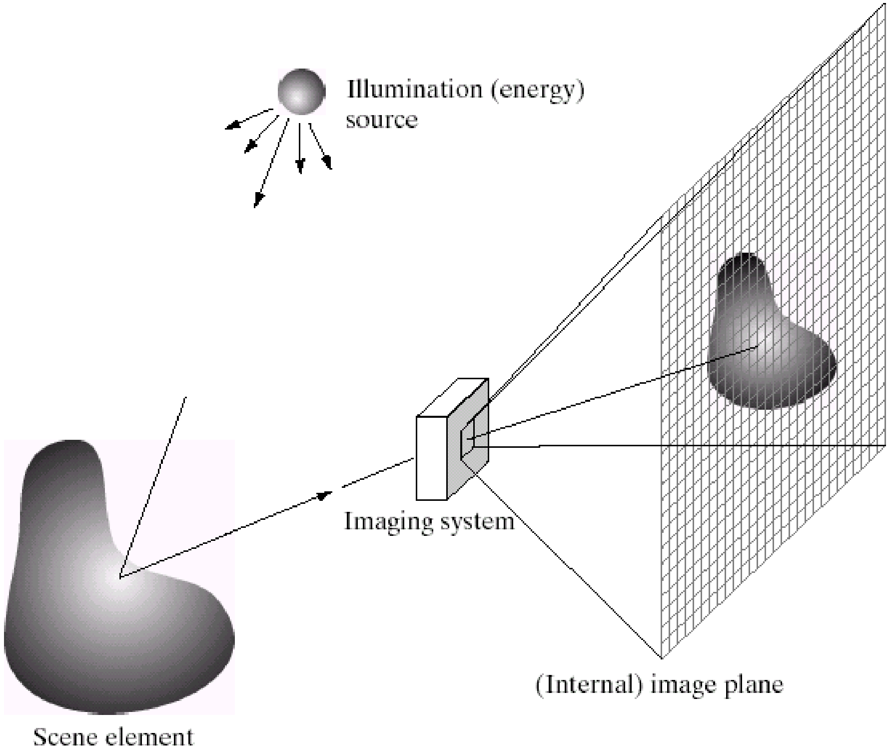


Image Acquisition

Images are typically generated by *illuminating* a *scene* and absorbing the energy reflected by the objects in that scene



- Typical notions of illumination and scene can be way off:
  - X-rays of a skeleton
  - Ultrasound of an unborn baby
  - Electro-microscopic images of molecules

Image Sampling and Quantisation

A digital sensor can only measure a limited number of **samples** at a **discrete** set of energy levels

Quantisation is the process of converting a continuous **analogue** signal into a digital representation of this signal

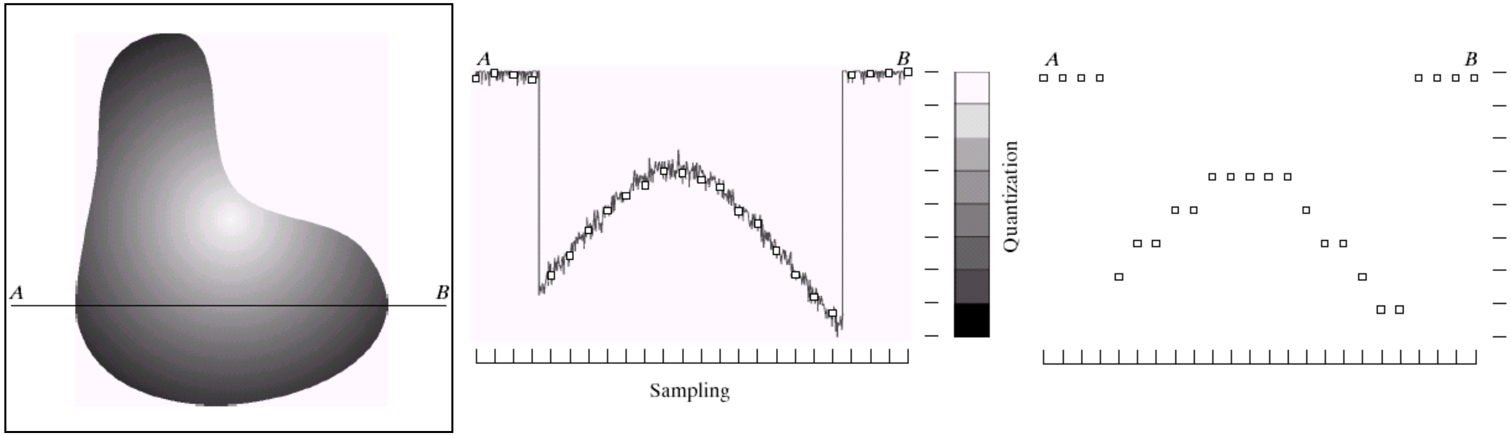
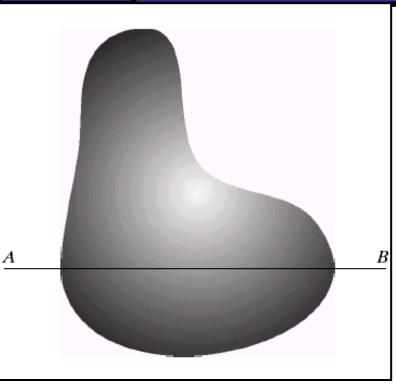


Image Sampling and Quantisation



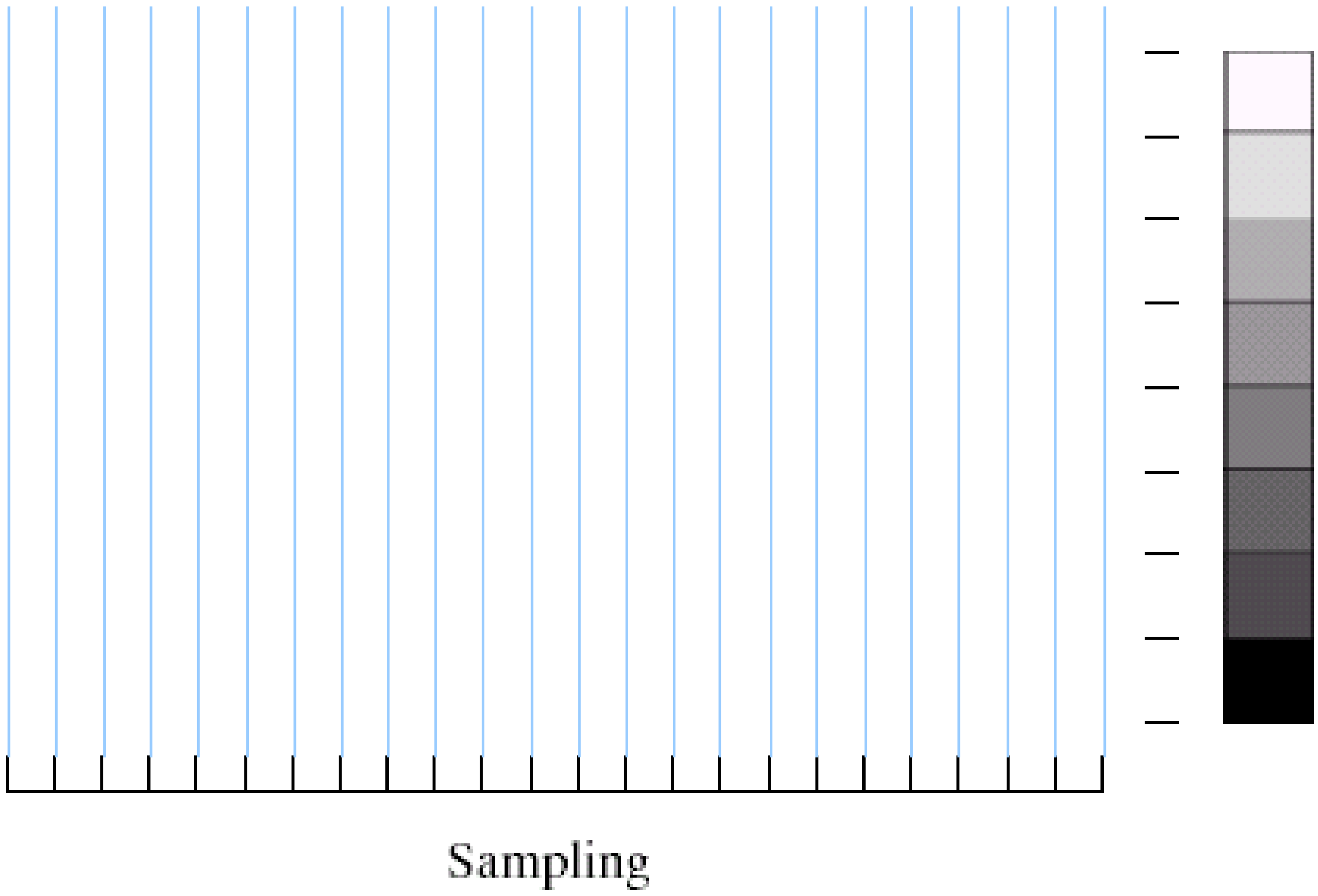
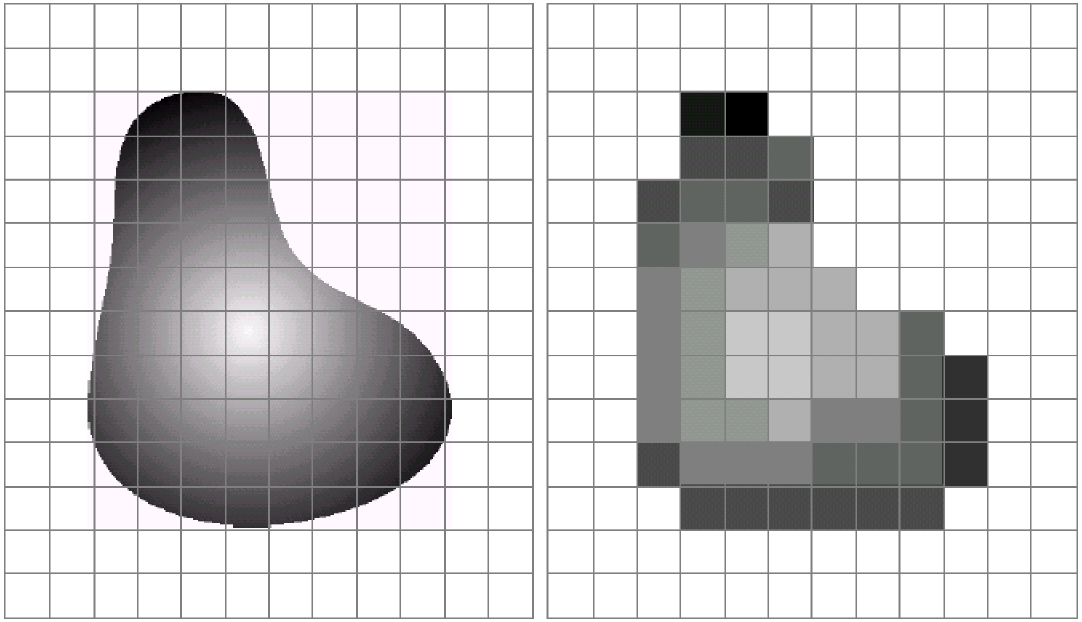


Image Sampling and Quantisation (cont…)

Remember that a digital image is always only an **approximation** of a real world scene



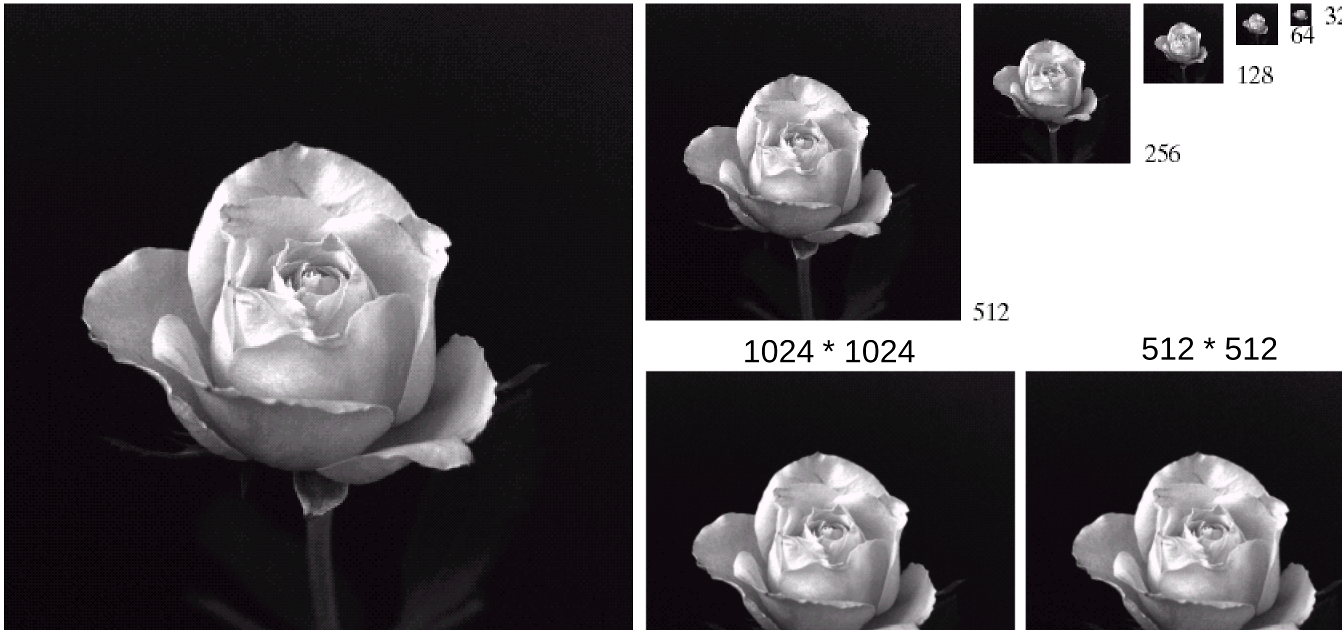
Spatial Resolution

The spatial resolution of an image is determined by how sampling was carried out

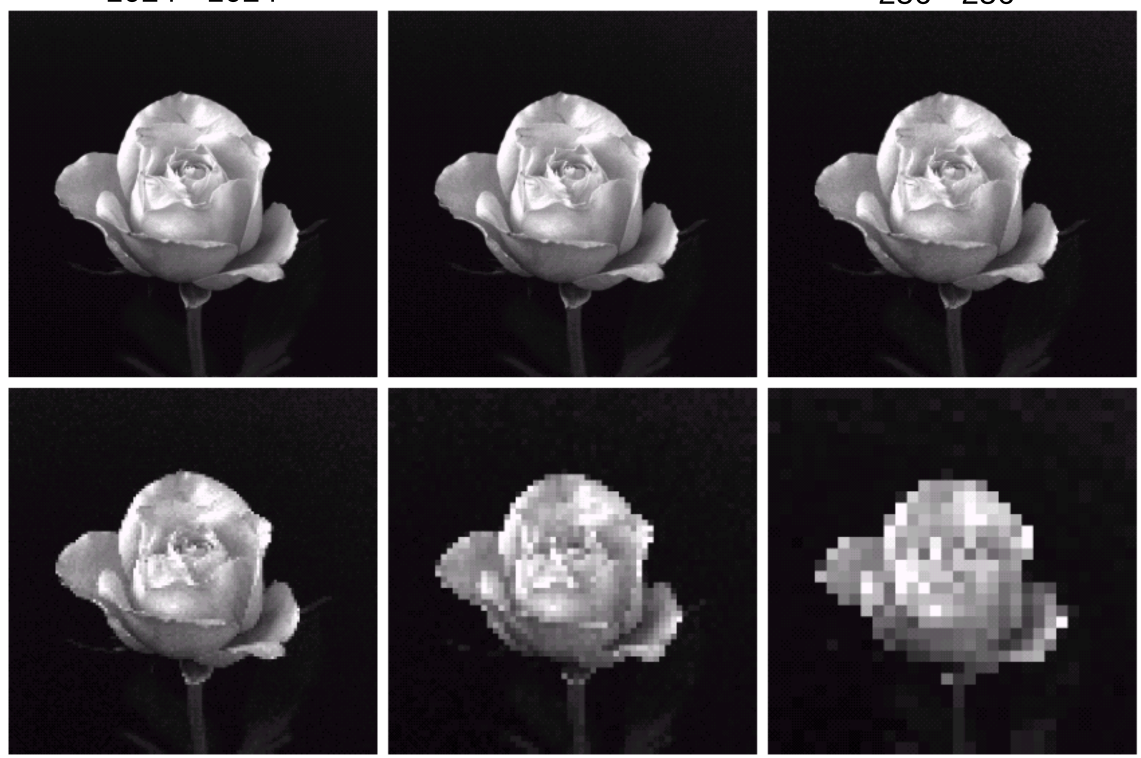
Spatial resolution simply refers to the smallest discernable detail in an image

- Vision specialists will often talk about pixel size
- Graphic designers will talk about *dots per inch* (DPI)

Spatial Resolution (cont…)



1024 * 1024 512 * 512 256 * 256



128 * 128 64 * 64 32 * 32

Intensity Level Resolution

Intensity level resolution refers to the number of intensity levels used to represent the image

- The more intensity levels used, the finer the level of detail discernable in an image
- Intensity level resolution is usually given in terms of the number of bits used to store each intensity level

Number of Bits	**Number of Intensity Levels**	**Examples**
1	2	0, 1
2	4	00, 01, 10, 11
4	16	0000, 0101, 1111
8	256	00110011, 01010101
16	65,536	1010101010101010

Intensity Level Resolution (cont…)

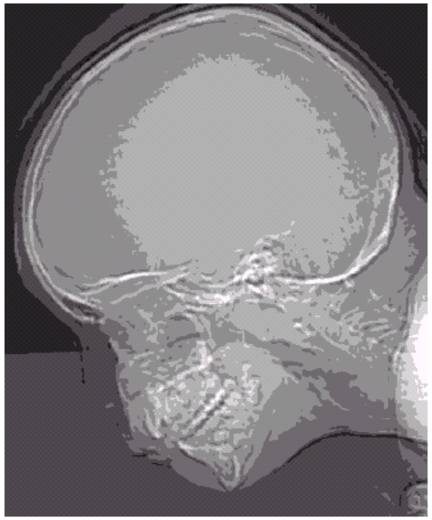
256 grey levels (8 bits per pixel)

128 grey levels (7 bpp)

64 grey levels (6 bpp)

32 grey levels (5 bpp)

16 grey levels (4 bpp)



8 grey levels (3 bpp)

4 grey levels (2 bpp)

2 grey levels (1 bpp)

Resolution: How Much Is Enough?

The big question with resolution is always *how much is enough*?

- This all depends on what is in the image and what you would like to do with it
- Key questions include
  - Does the image look aesthetically pleasing?
  - Can you see what you need to see within the image?

Resolution: How Much Is Enough? (cont…)

The picture on the right is fine for counting the number of cars, but not for reading the number plate

Summary

We have looked at:

– Human visual system

– Light and the electromagnetic spectrum

– Image representation

– Image sensing and acquisition

– Sampling, quantisation and resolution

Next time we start to look at techniques for image enhancement

Digital Image Processing

Image Enhancement

(Histogram Processing)

Contents

Over the next few lectures we will look at image enhancement techniques working in the spatial domain:

– What is image enhancement?
– Different kinds of image enhancement
– Histogram processing
– Point processing
– Neighbourhood operations

A Note About Grey Levels

So far when we have spoken about image grey level values we have said they are in the range [0, 255]

- Where 0 is black and 255 is white

There is no reason why we have to use this range

- The range [0,255] stems from display

For many of the image processing operations in this lecture grey levels are assumed to be given in the range [0.0, 1.0]

What Is Image Enhancement?

Image enhancement is the process of making images more useful

The reasons for doing this include:

- Highlighting interesting detail in images
- Removing noise from images
- Making images more visually appealing

Image Enhancement Examples

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Spatial & Frequency Domains

There are two broad categories of image enhancement techniques

- Spatial domain techniques
  - Direct manipulation (تعامل) of image pixels
- Frequency domain techniques
  - Manipulation (تعامل) of Fourier transform or wavelet transform of an image

For the moment we will concentrate on techniques that operate in the spatial domain

Background

the spatial domain process is denoted by

$$g(x,y) = T[f(x,y)]$$

$f(x,y)$ = i/p image $\quad\quad$ $g(x,y)$= o/p image

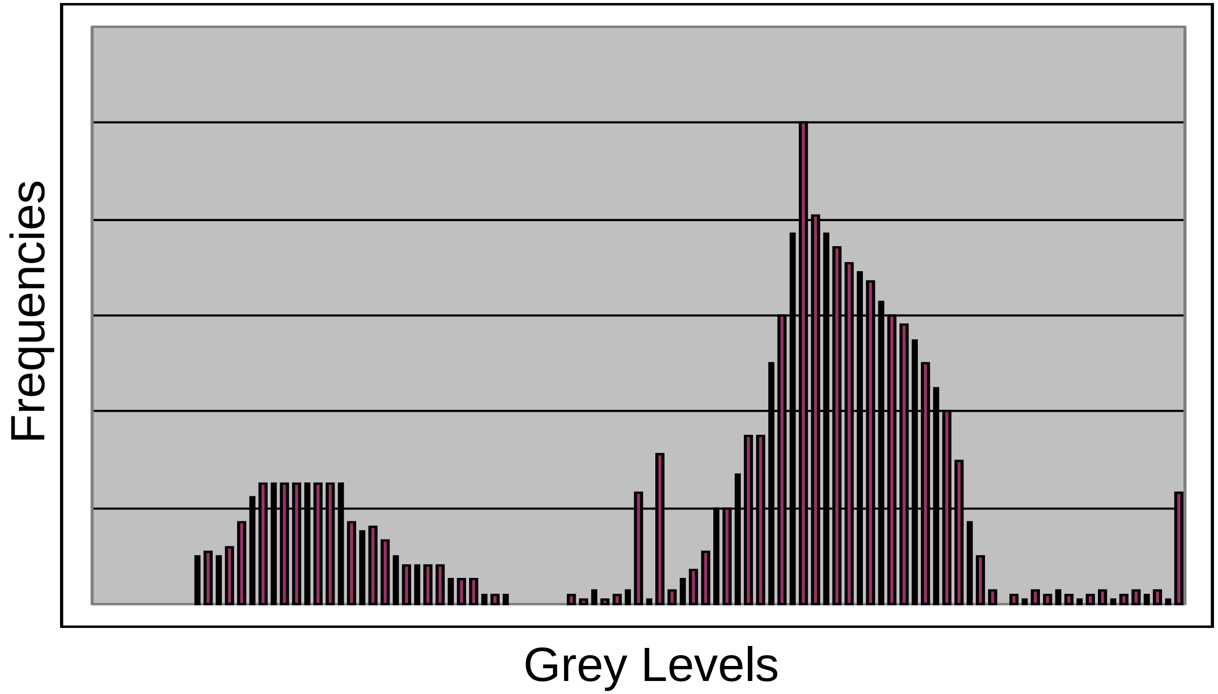
T = operator on f

The value of g at (x,y) depends only on the intensity of f at that point.

Image Histograms

The histogram of an image shows us the distribution of grey levels in the image

Massively useful in image processing, especially in segmentation



Histogram(cont.)

Histogram is defined by $h(r_k)=n_k$

r_k kth intensity level in the interval[0 G]

n_k the no. of pixels in the image whose intensity level is r_k

Normalized histogram $P(r_k) = h(r_k)/n = n_k/n$

$k=1\ldots\ldots L$ where $L = G+1$

H = imhist(f,b)

b= no. of bins if not included default =256

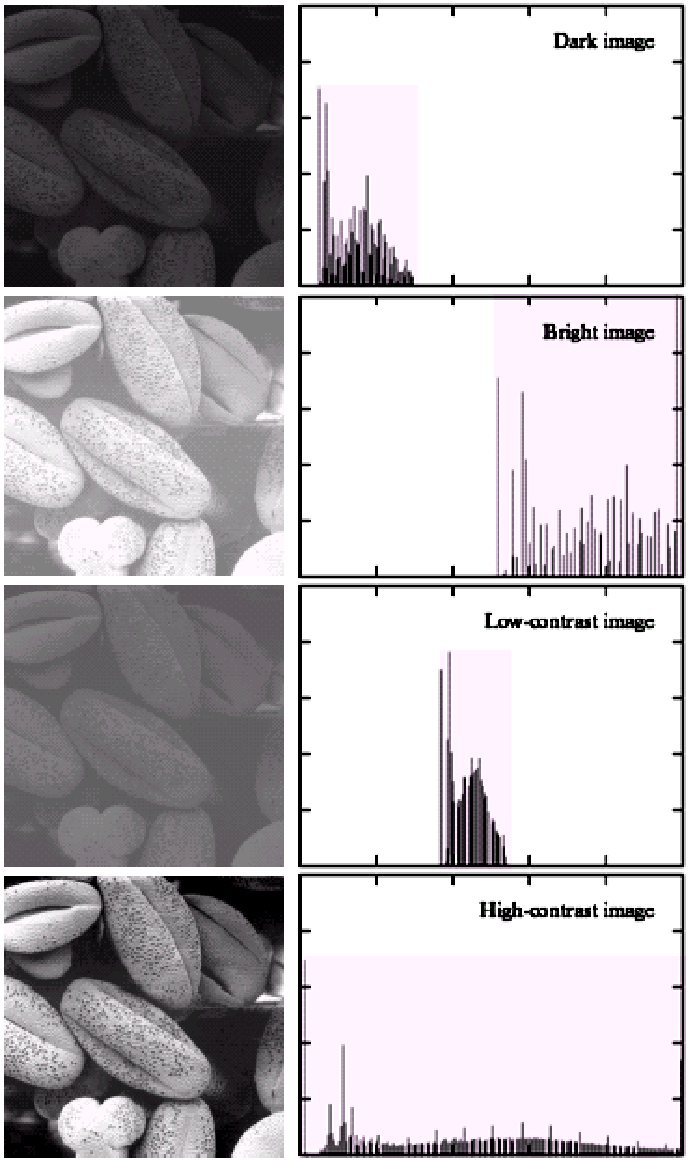
Histogram Examples (cont…)

A selection of images and their histograms

Notice the relationships between the images and their histograms

Note that the high contrast image has the most evenly spaced histogram

Enhancement is achieved by spreading the
Levels of the input image over a wider rang
Of the intensity scale.



Histogram Equalisation

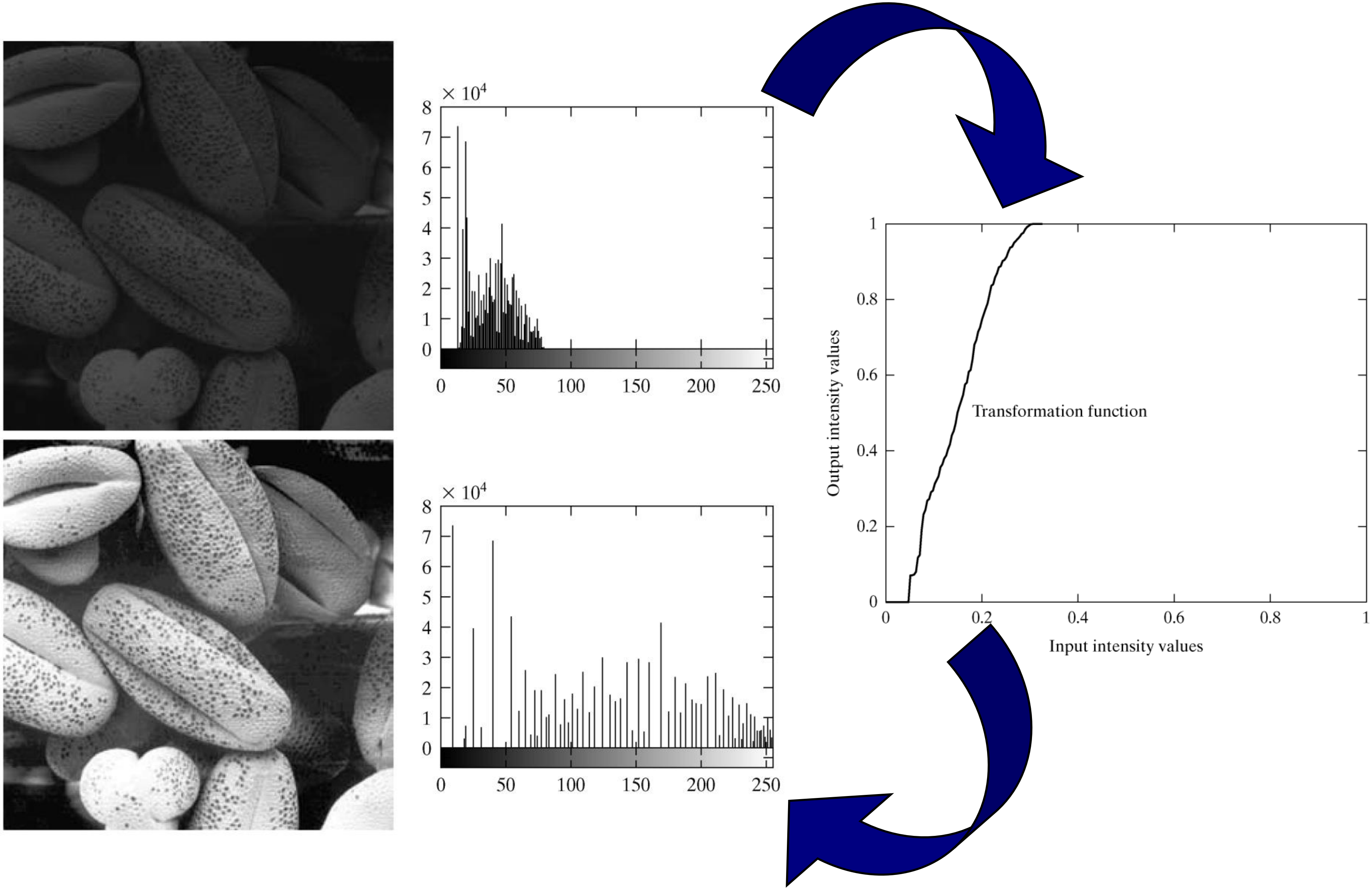
Spreading out the frequencies in an image (or equalising the image) is a simple way to improve dark or washed out images

The formula for histogram equalisation is given where

- r_k: input intensity
- s_k: processed intensity
- k: the intensity range (e.g 0.0 – 1.0)
- n_j: the frequency of intensity j
- n: the sum of all frequencies

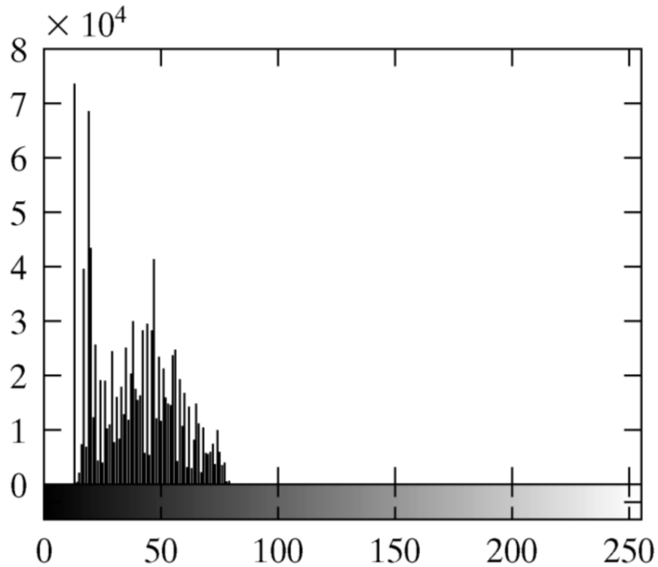
$$\begin{aligned} s_k &= T(r_k) \\ &= \sum_{j=1}^{k} p_r(r_j) \\ &= \sum_{j=1}^{k} \frac{n_j}{n} \end{aligned}$$

Equalisation Transformation Function

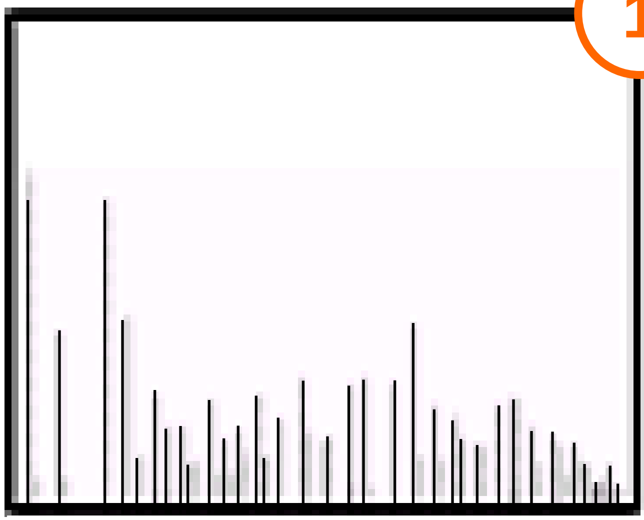


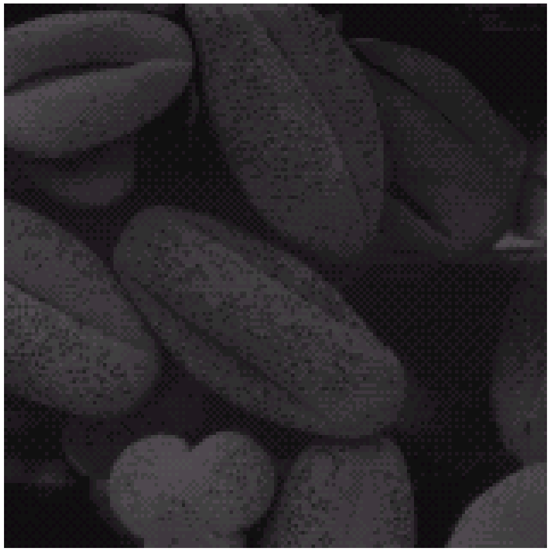
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Equalisation Examples



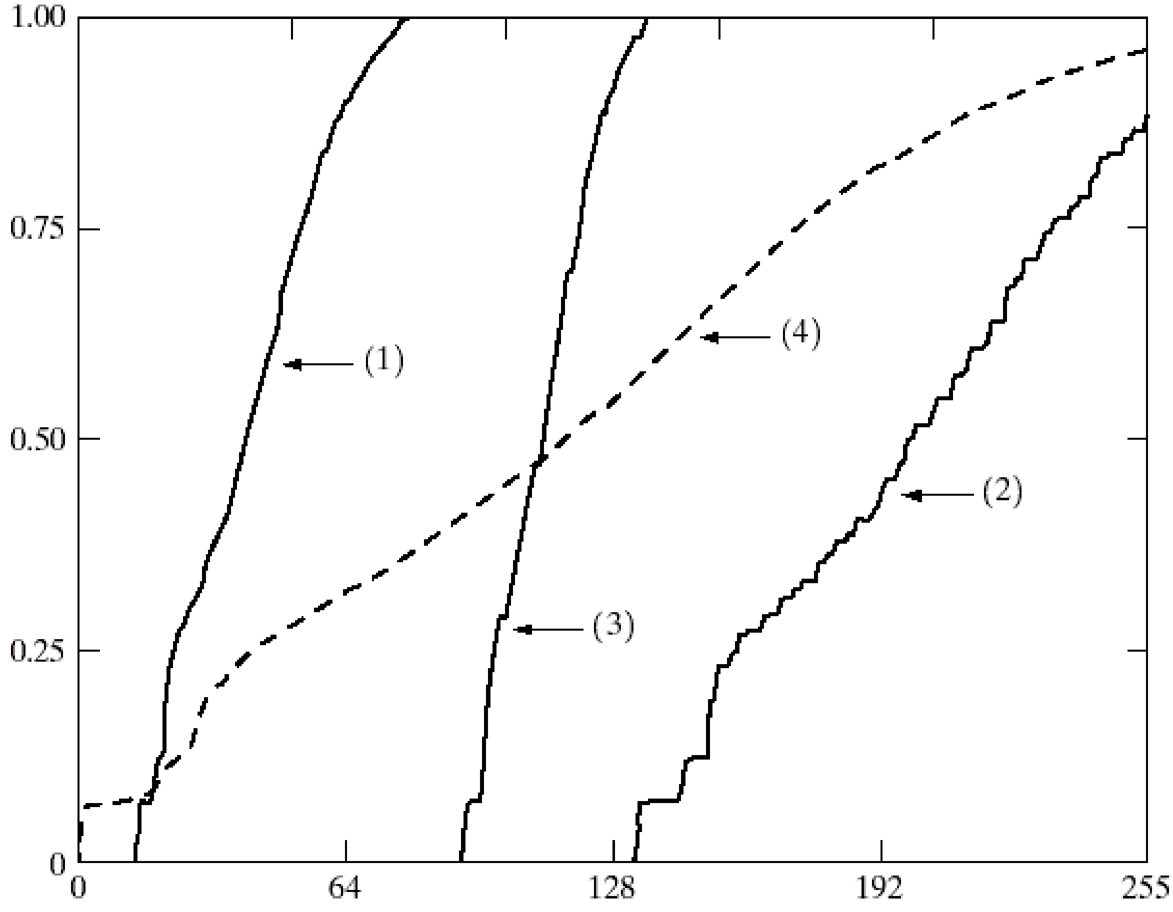
1



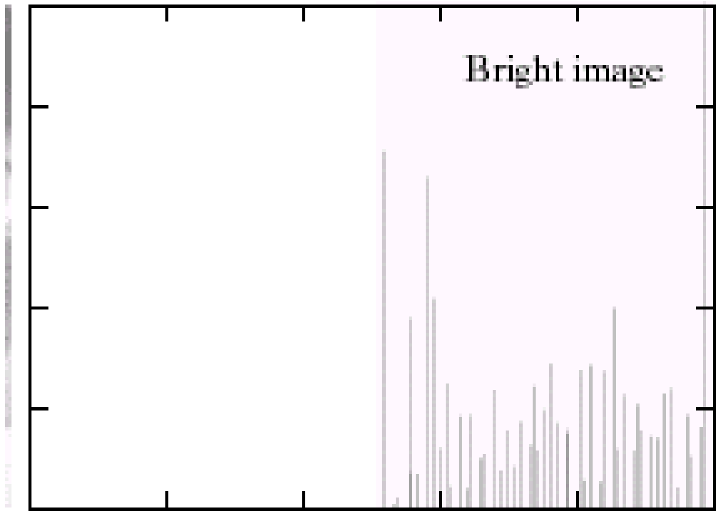


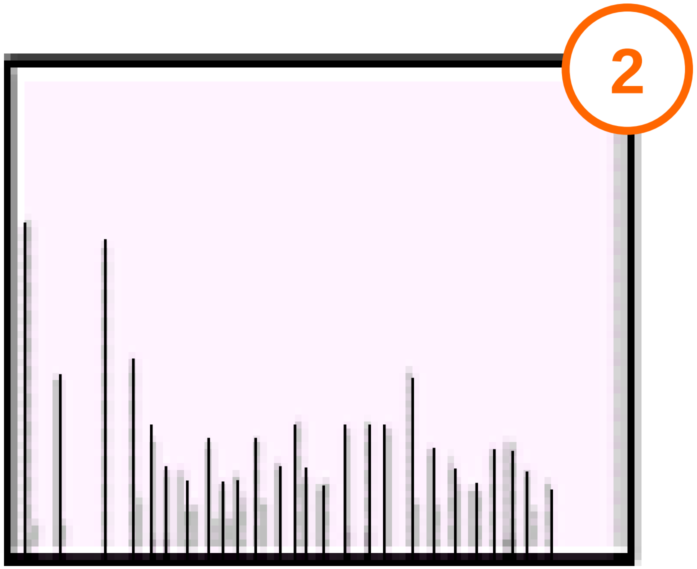
Equalisation Transformation Functions

The functions used to equalise the images in the previous example

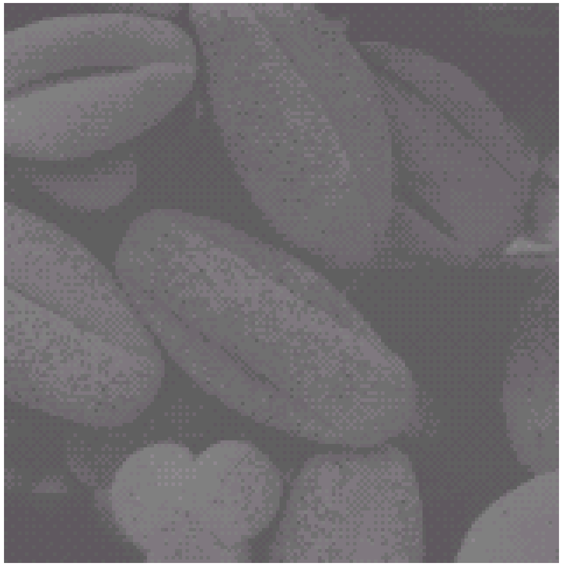


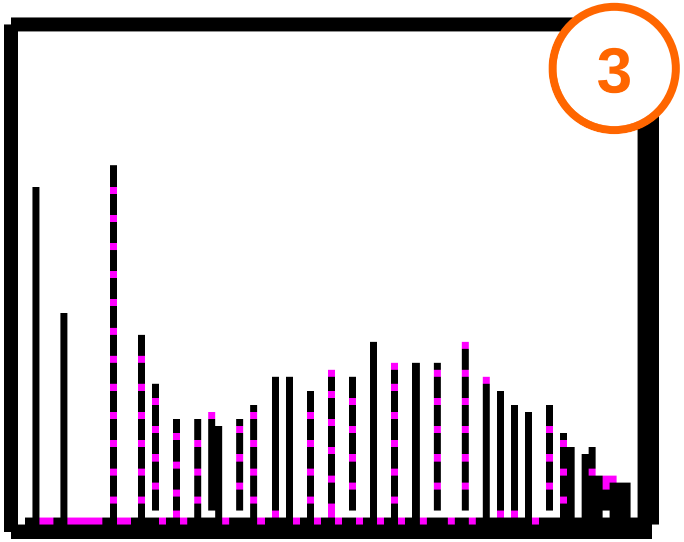
Equalisation Examples



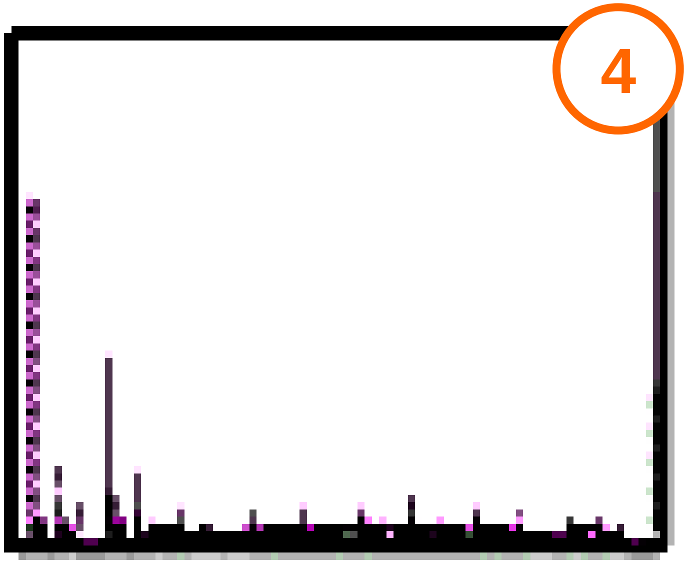


Equalisation Examples (cont…)



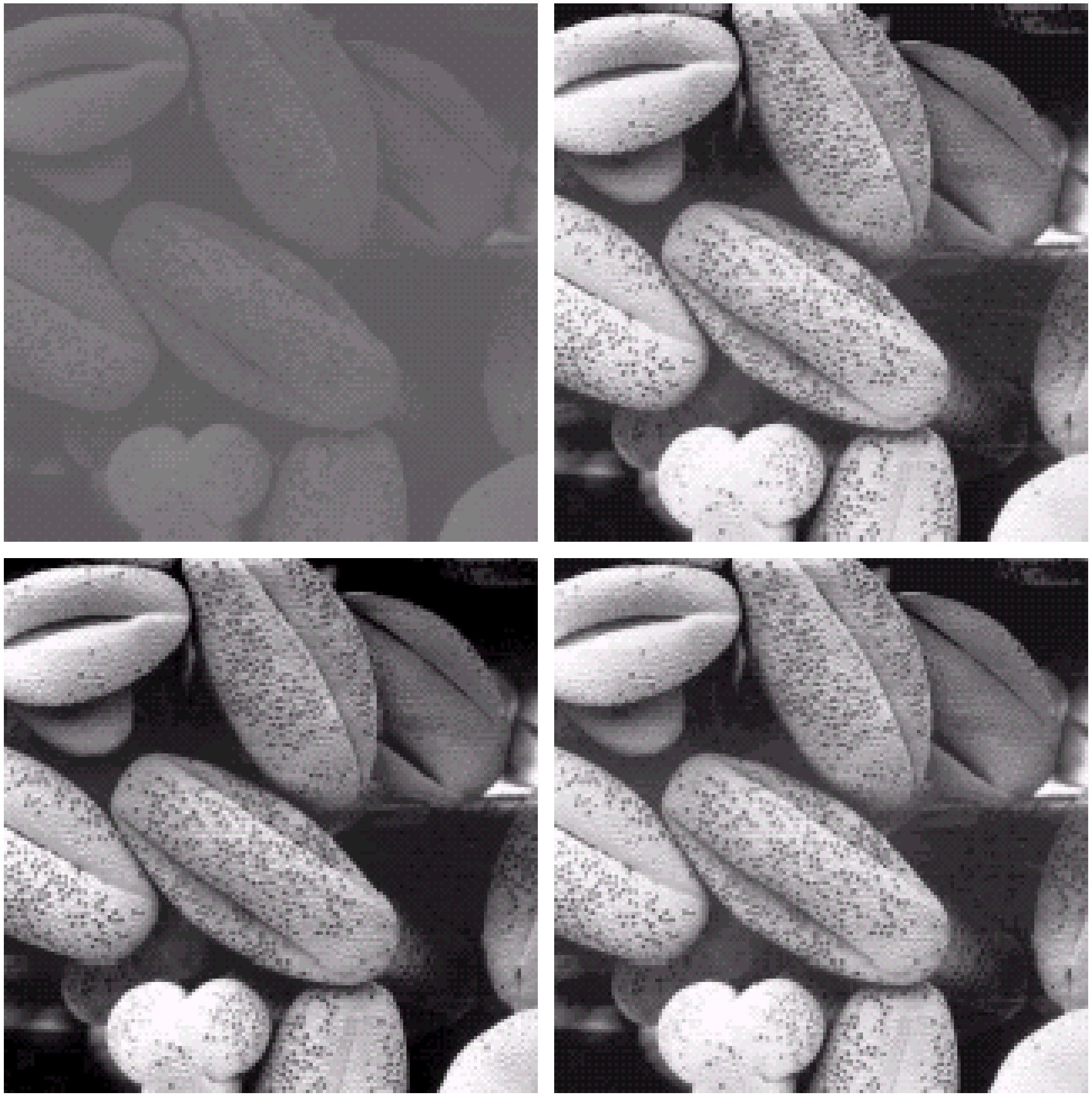


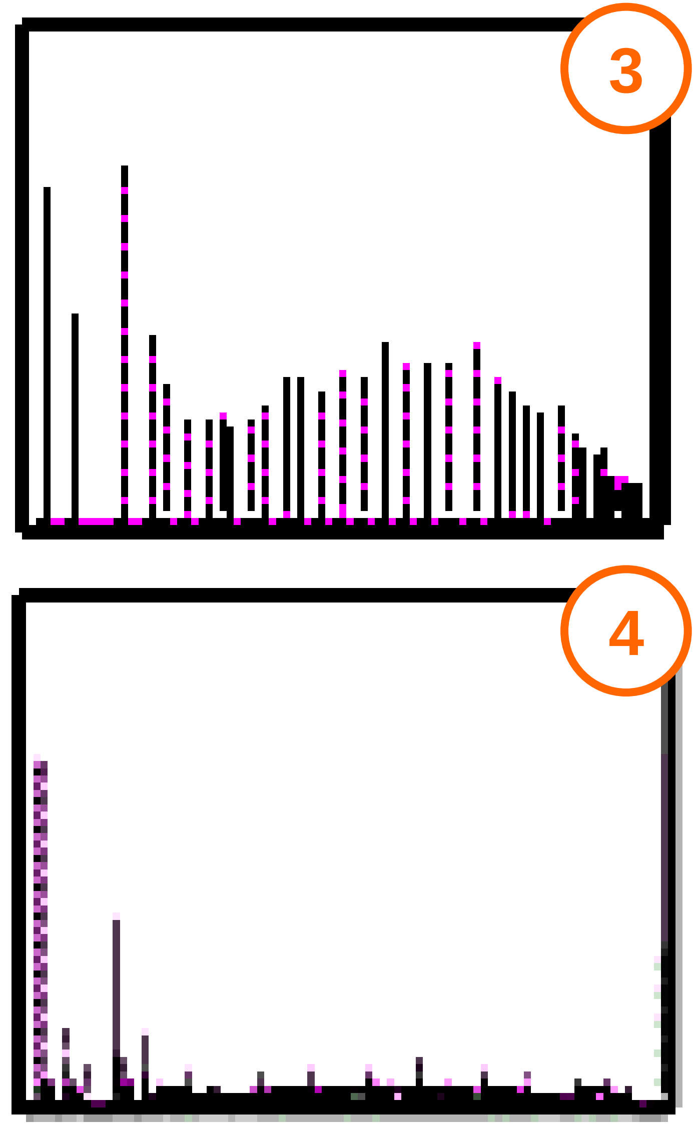




Equalisation Examples (cont…)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)





Summary

We have looked at:

– Different kinds of image enhancement

– Histograms

– Histogram equalisation

Next time we will start to look at point processing and some neighbourhood operations

Digital Image Processing

Image Enhancement

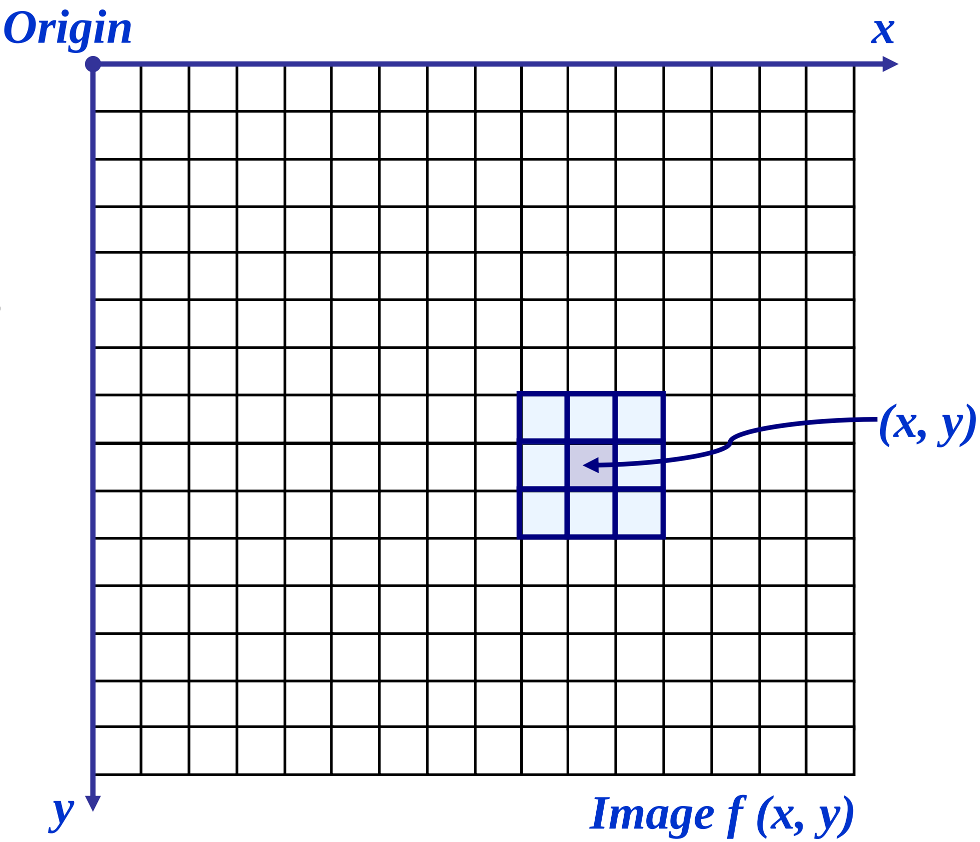
(Point Processing)

Basic Spatial Domain Image Enhancement

Most spatial domain enhancement operations can be reduced to the form

$g(x, y) = T[f(x, y)]$

where $f(x, y)$ is the input image, $g(x, y)$ is the processed image and T is some operator defined over some neighbourhood of (x, y)



Point Processing

The simplest spatial domain operations occur when the neighbourhood is simply the pixel itself

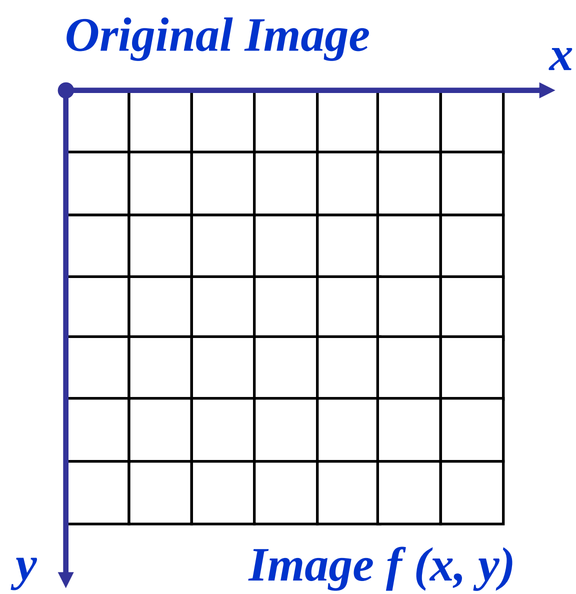
In this case T is referred to as a *grey level transformation function* or a *point processing operation*

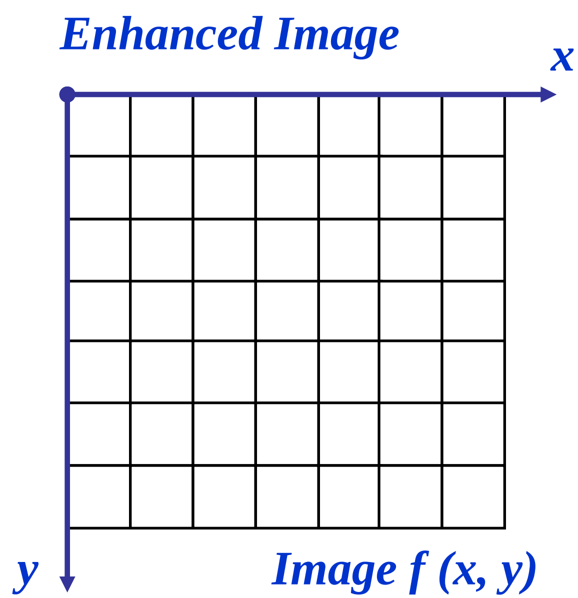
Point processing operations take the form

$$s = T(r)$$

where s refers to the processed image pixel value and r refers to the original image pixel value

Point Processing Example: Negative Images (cont…)



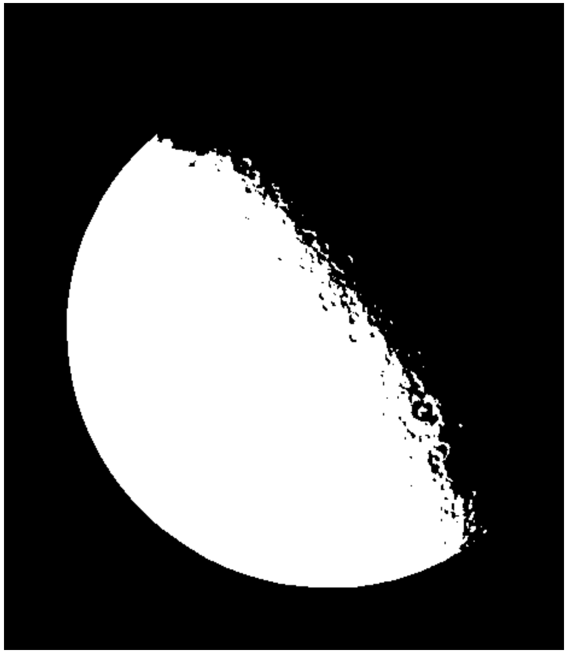


$$s = intensity_{max} - r$$

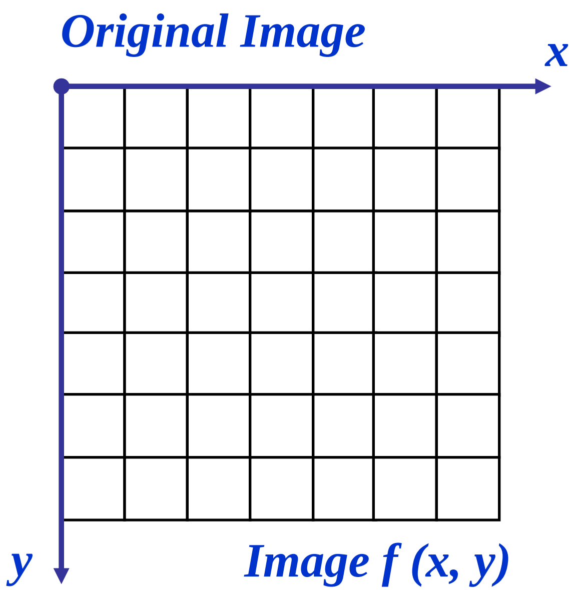
Point Processing Example: Thresholding

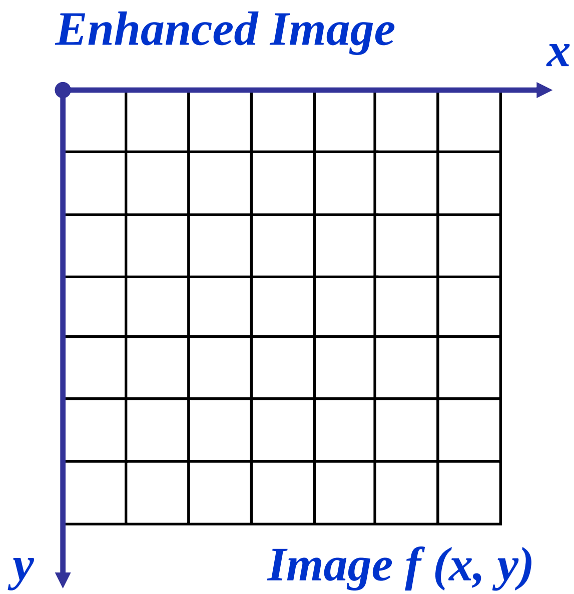
Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background

$$s = \begin{cases} 1.0 & r > \text{threshold} \\ 0.0 & r <= \text{threshold} \end{cases}$$



Point Processing Example: Thresholding (cont…)





$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

Digital Image Processing

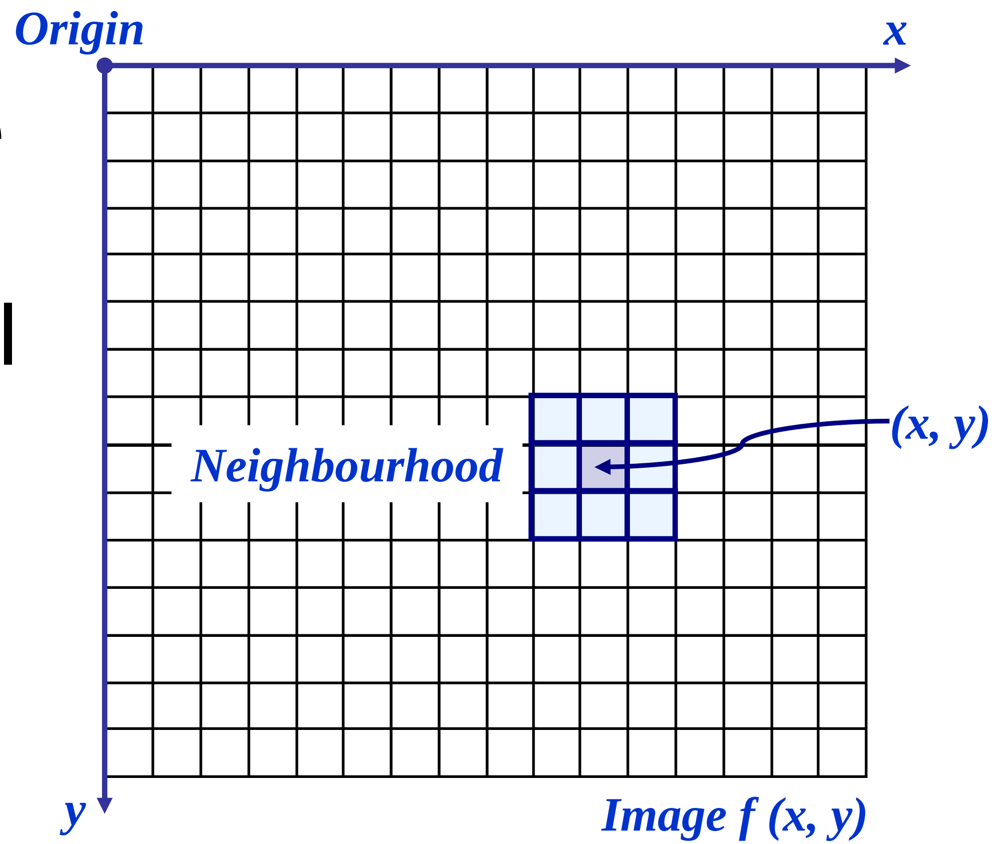
- Image Enhancement (Spatial Filtering 1)

Contents

- In this lecture we will look at spatial filtering techniques:
  - Neighbourhood operations
  - What is spatial filtering?
  - Smoothing operations
  - What happens at the edges?
  - Correlation and convolution

Neighbourhood Operations

- Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations
- Neighbourhoods are mostly a rectangle around a central pixel
- Any size rectangle and any shape filter are possible



Simple Neighbourhood Operations

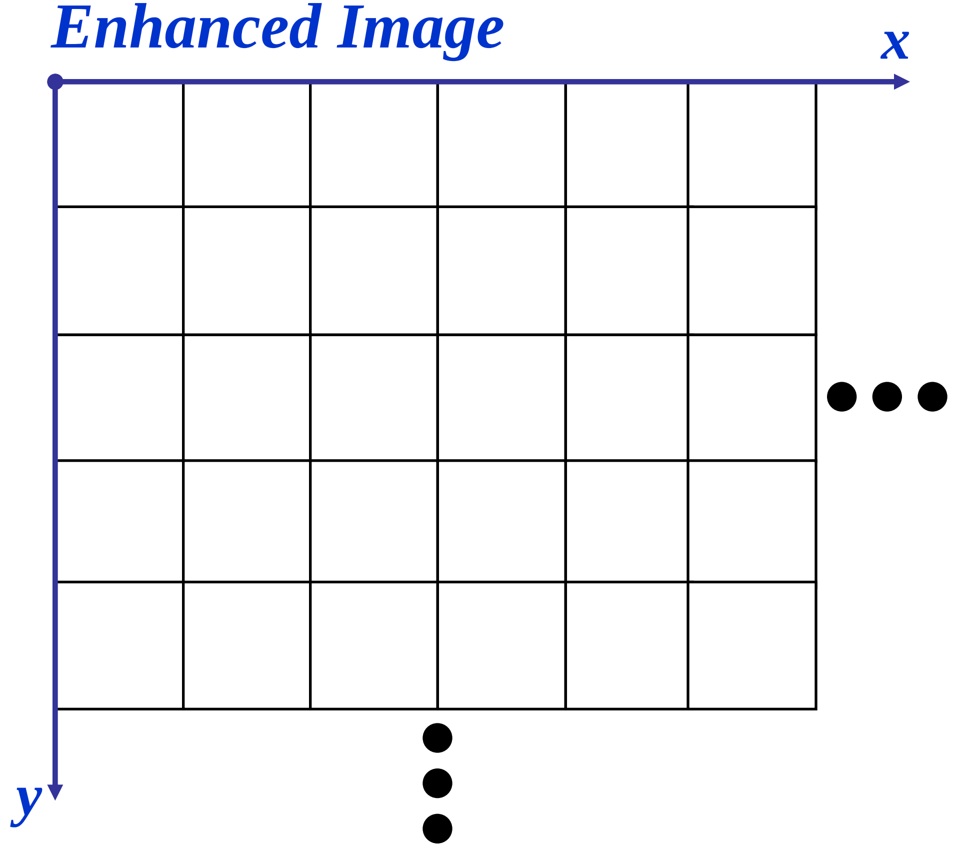
- Some simple neighbourhood operations include:
  - **Min:** Set the pixel value to the minimum in the neighbourhood
  - **Max:** Set the pixel value to the maximum in the neighbourhood
  - **Median:** The median value of a set of numbers is the midpoint value in that set (e.g. from the set [1, 7, 15, 18, 24] 15 is the median). Sometimes the median works better than the average

Simple Neighbourhood Operations Example

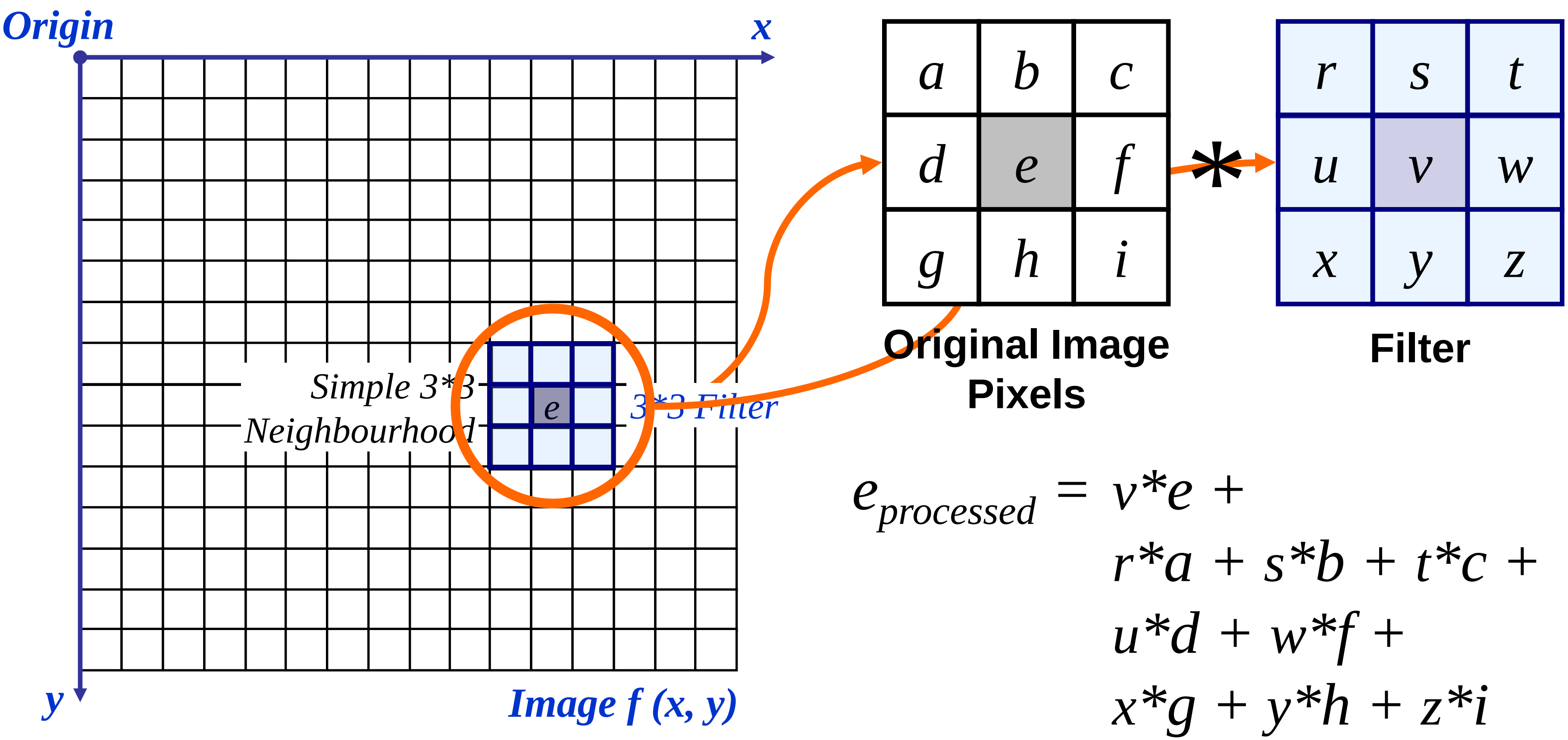
Original Image

123	127	128	119	115	130
140	145	148	153	167	172
133	154	183	192	194	191
194	199	207	210	198	195
164	170	175	162	173	151

Enhanced Image



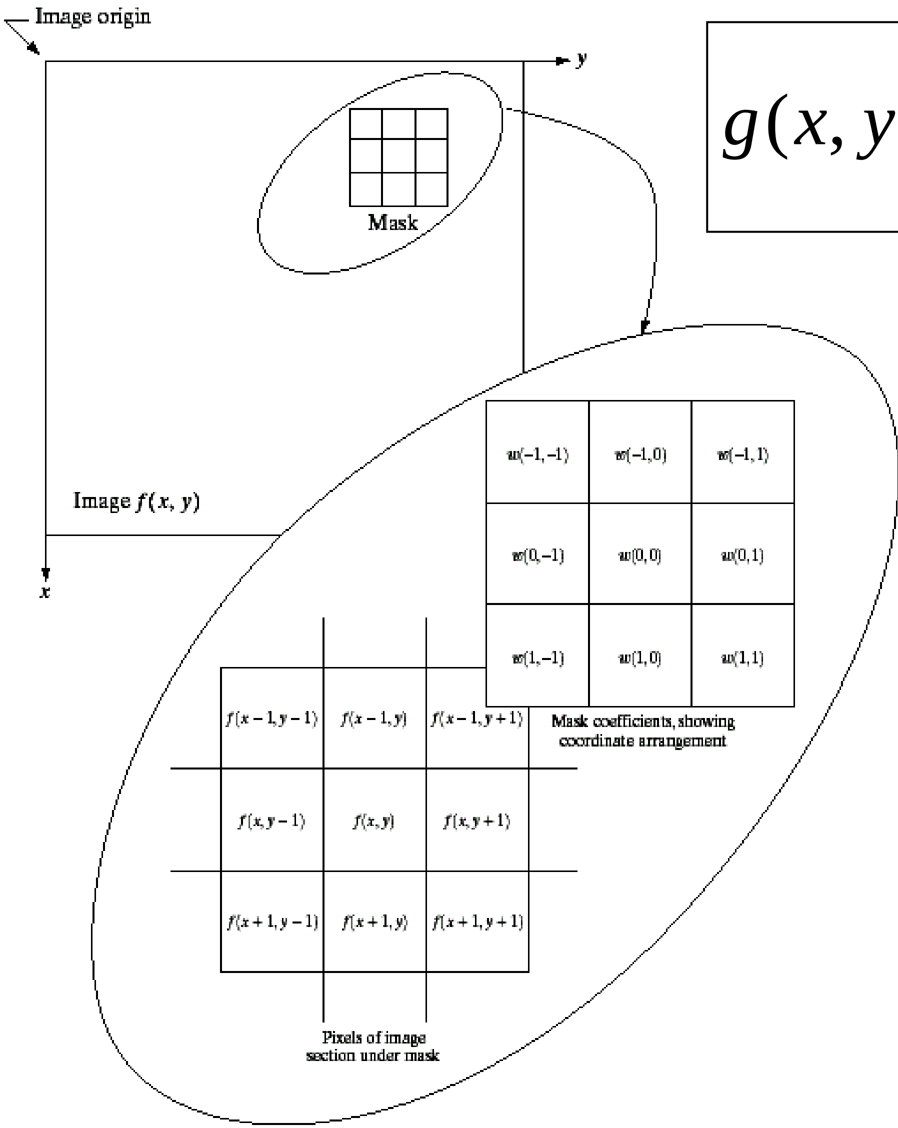
The Spatial Filtering Process



$$e_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

The above is repeated for every pixel in the original image to generate the smoothed image

Spatial Filtering: Equation Form



$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

Filtering can be given in equation form as shown above

Notations are based on the image shown to the left

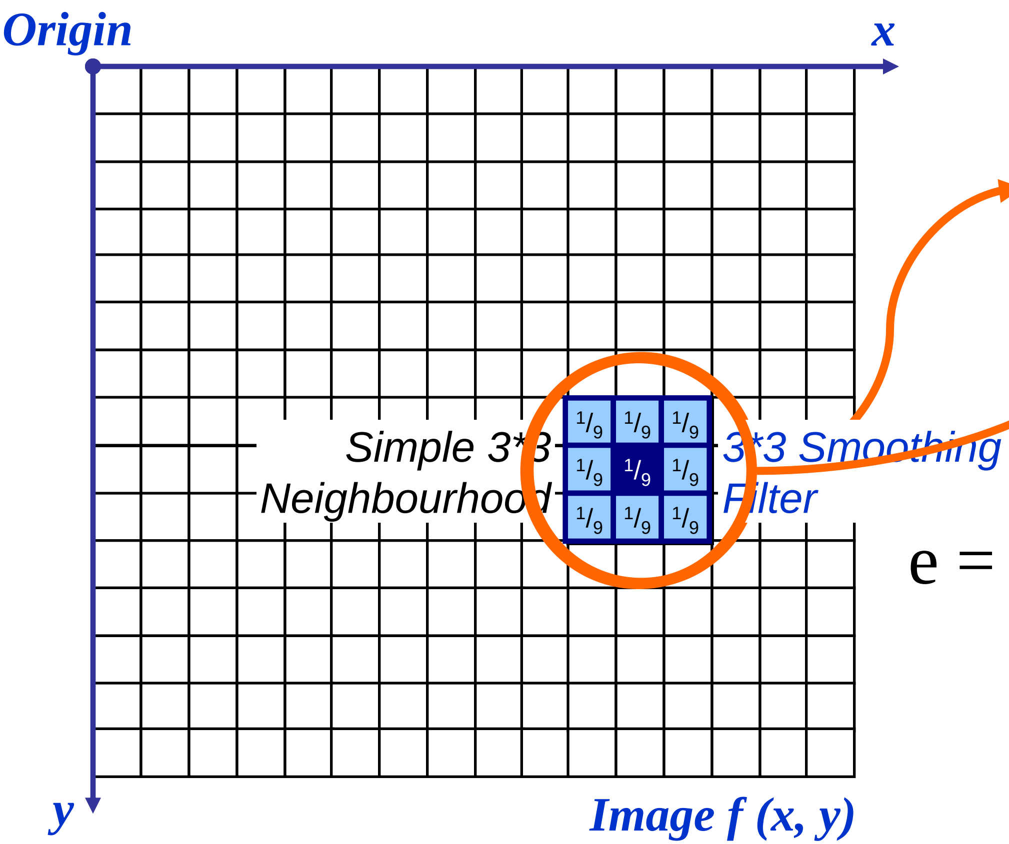
Smoothing Spatial Filters

- One of the simplest spatial filtering operations we can perform is a smoothing operation
  - Simply average all of the pixels in a neighbourhood around a central value
  - Especially useful in removing noise from images
  - Also useful for highlighting gross detail

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Simple averaging filter

Smoothing Spatial Filtering



104	100	108
99	106	98
95	90	85

Original Image Pixels

$*$

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

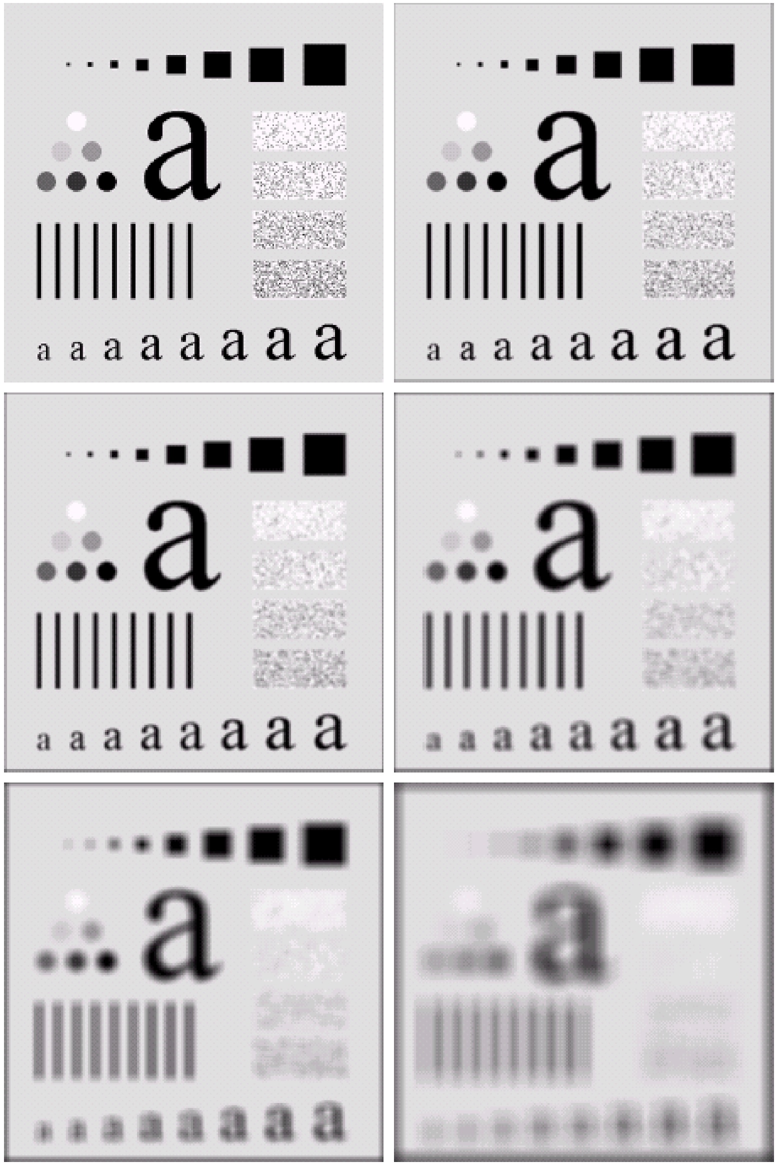
Filter

$$
\begin{aligned}
e = {} & {}^1/_9 * 106 + \\
& {}^1/_9 * 104 + {}^1/_9 * 100 + {}^1/_9 * 108 + \\
& {}^1/_9 * 99 + {}^1/_9 * 98 + \\
& {}^1/_9 * 95 + {}^1/_9 * 90 + {}^1/_9 * 85 \\
= {} & 98.3333
\end{aligned}
$$

The above is repeated for every pixel in the original image to generate the smoothed image

Image Smoothing Example

- The image at the top left is an original image of size 500*500 pixels
- The subsequent images show the image after filtering with an averaging filter of increasing sizes
  - 3, 5, 9, 15 and 35
- Notice how detail begins to disappear



Weighted Smoothing Filters

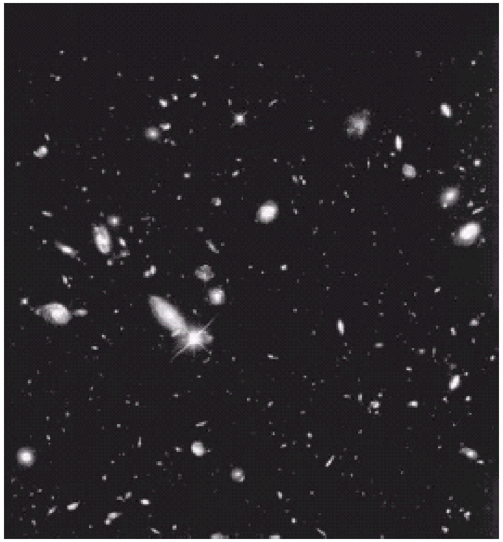
- More effective smoothing filters can be generated by allowing different pixels in the neighbourhood different weights in the averaging function
  - Pixels closer to the central pixel are more important
  - Often referred to as a *weighted averaging*

$^1/_{16}$	$^2/_{16}$	$^1/_{16}$
$^2/_{16}$	$^4/_{16}$	$^2/_{16}$
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$

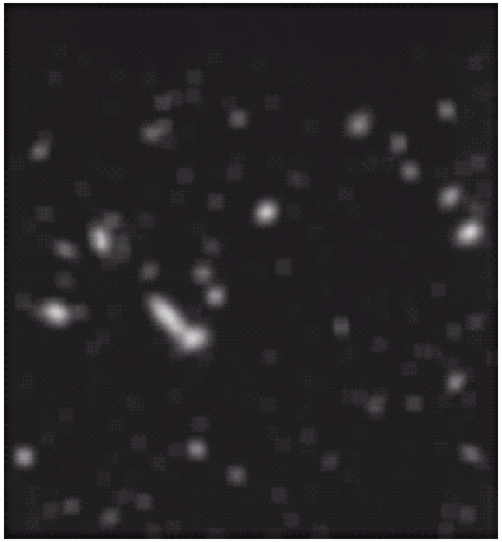
Weighted averaging filter

Another Smoothing Example

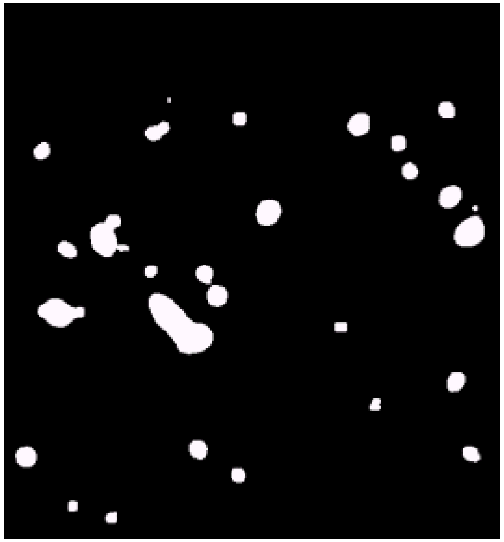
- By smoothing the original image we get rid of lots of the finer detail which leaves only the gross features for thresholding



Original Image

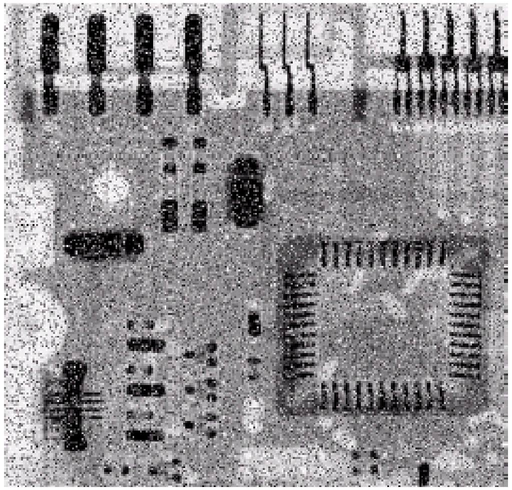


Smoothed Image



Thresholded Image

Averaging Filter Vs. Median Filter Example



Original Image With Noise

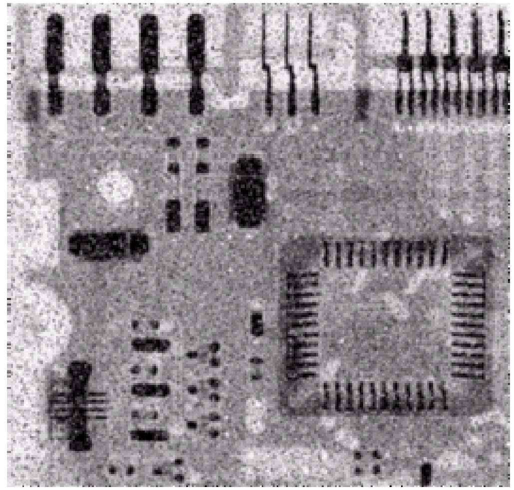


Image After Averaging Filter

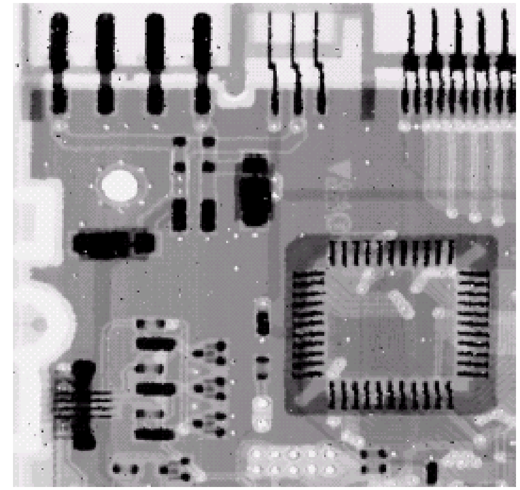
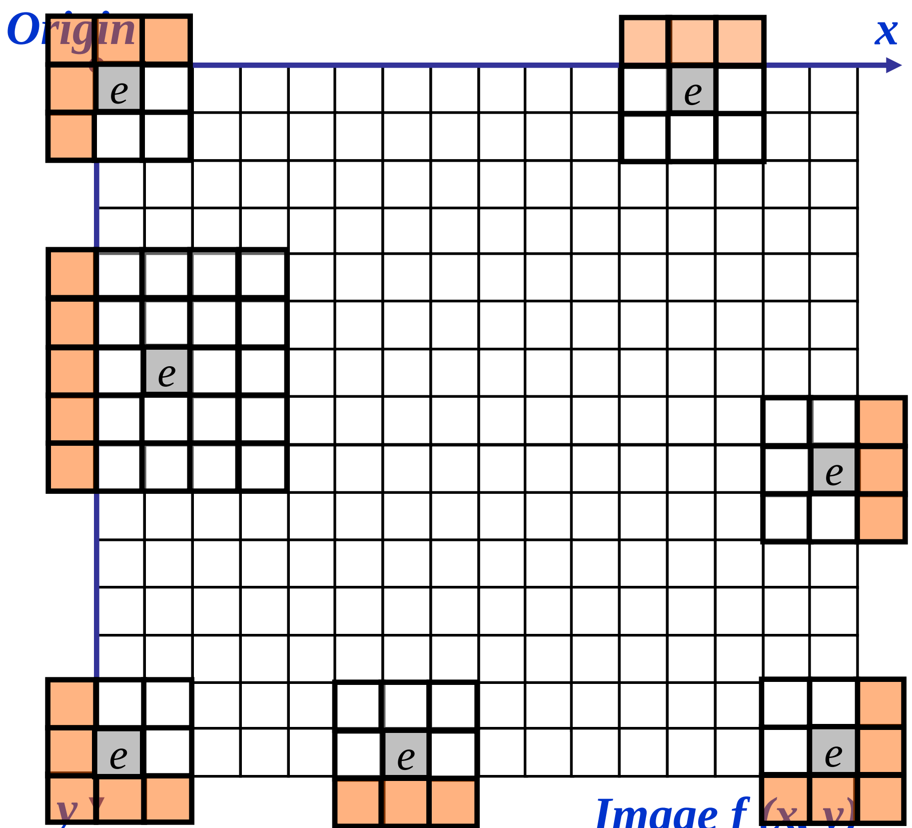


Image After Median Filter

- Filtering is often used to remove noise from images
- Sometimes a median filter works better than an averaging filter

Strange Things Happen At The Edges!

At the edges of an image we are missing pixels to form a neighbourhood



Strange Things Happen At The Edges! (cont…)

- There are a few approaches to deal with missing edge pixels:
  - Omit(يحذف) missing pixels
    - Only works with some filters
    - Can add extra code and slow down processing
  - Pad(يحشو) the image
    - Typically with either all white or all black pixels
  - Replicate(يكرر/يضاعف) border pixels
  - Truncate(يبتر/يقلم) the image
  - Allow pixels *wrap (يلف)around* the image
    - Can cause some strange image artefacts

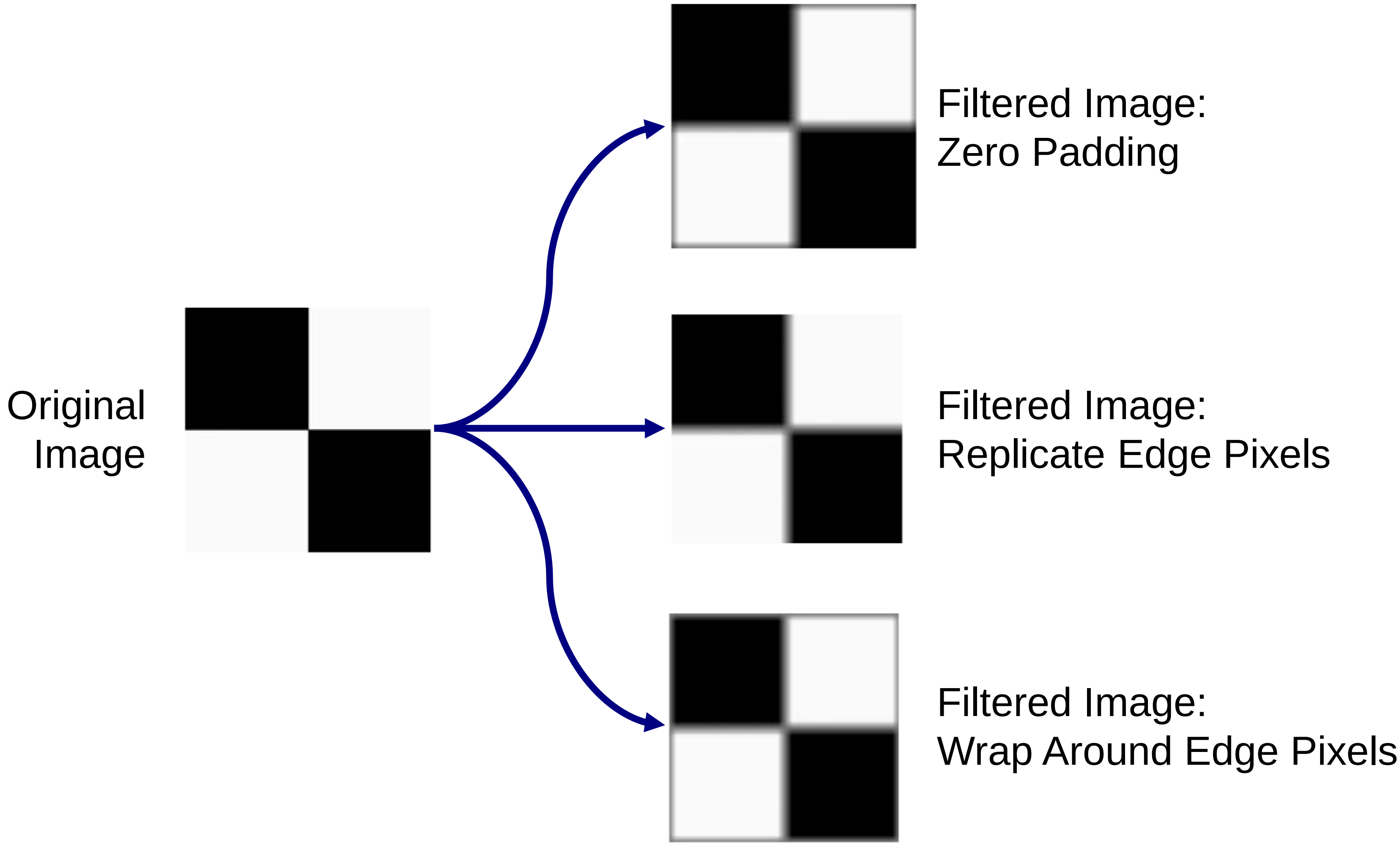
Simple Neighbourhood Operations Example

x

123	127	128	119	115	130
140	145	148	153	167	172
133	154	183	192	194	191
194	199	207	210	198	195
164	170	175	162	173	151

y

Strange Things Happen At The Edges! (cont…)



Correlation & Convolution

- The filtering we have been talking about so far is referred to as *correlation* with the filter itself referred to as the *correlation kernel*
- *Convolution* is a similar operation, with just one subtle difference

ROTATED BY 180°

a	b	c
d	e	e
f	g	h

Original Image Pixels

$*$

r	s	t
u	v	w
x	y	z

Filter

$$\begin{aligned} e_{processed} = {} & v*e + \\ & z*a + y*b + x*c + \\ & w*d + u*e + \\ & t*f + s*g + r*h \end{aligned}$$

- For symmetric filters it makes no difference

Summary

- In this lecture we have looked at the idea of spatial filtering and in particular:
  - Neighbourhood operations
  - The filtering process
  - Smoothing filters
  - Dealing with problems at image edges when using filtering
  - Correlation and convolution
- Next time we will looking at sharpening filters and more on filtering and image enhancement

Digital Image Processing

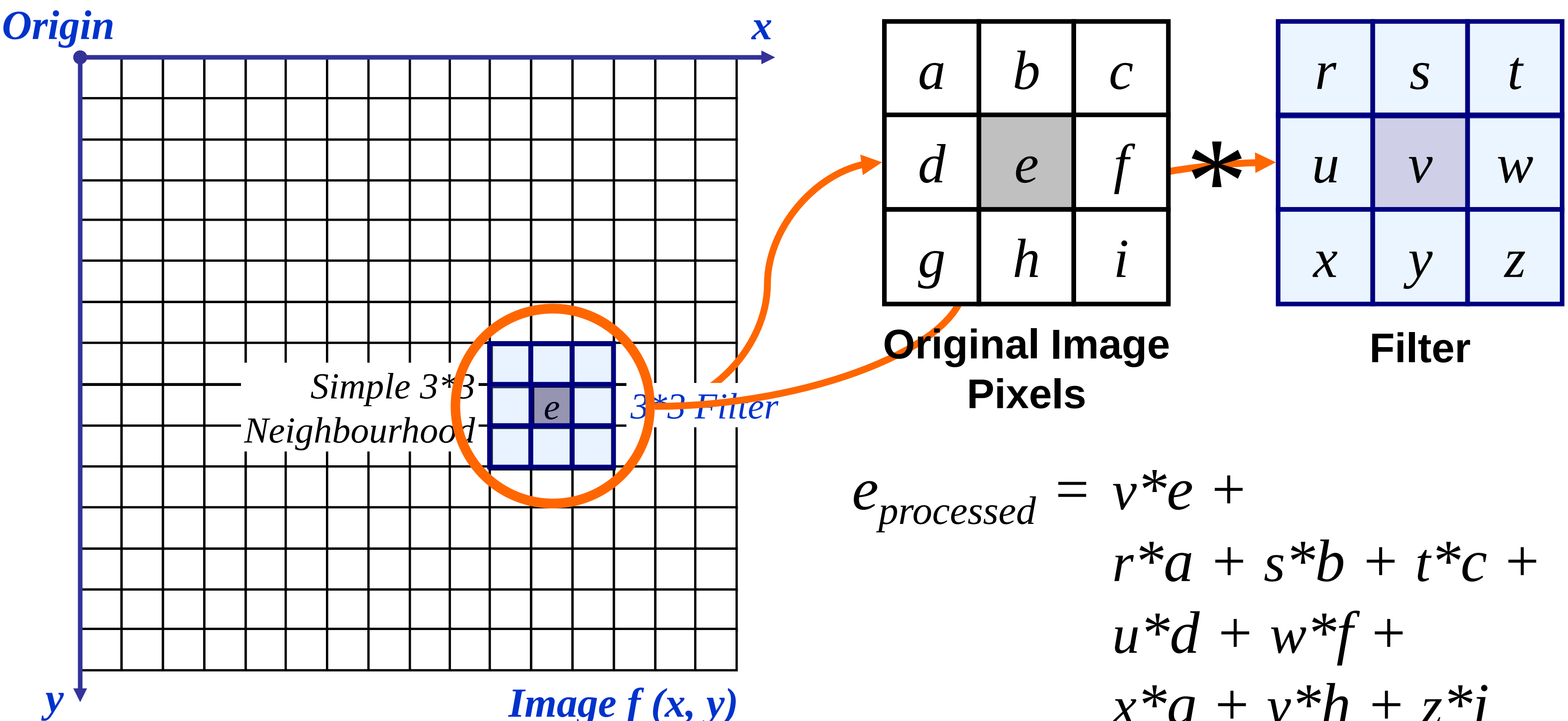
Image Enhancement
(Spatial Filtering 2)

Contents

In this lecture we will look at more spatial filtering techniques

- Spatial filtering refresher
- Sharpening filters
  - 1^{st} derivative filters
  - 2^{nd} derivative filters
- Combining filtering techniques

Spatial Filtering Refresher



$$e_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

The above is repeated for every pixel in the original image to generate the smoothed image

Sharpening Spatial Filters

Previously we have looked at smoothing filters which remove fine detail

Sharpening spatial filters seek to highlight fine detail

– Remove blurring from images

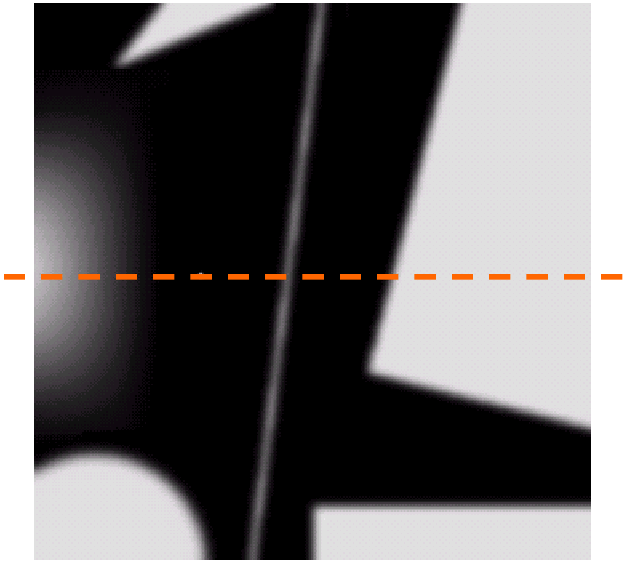
– Highlight edges

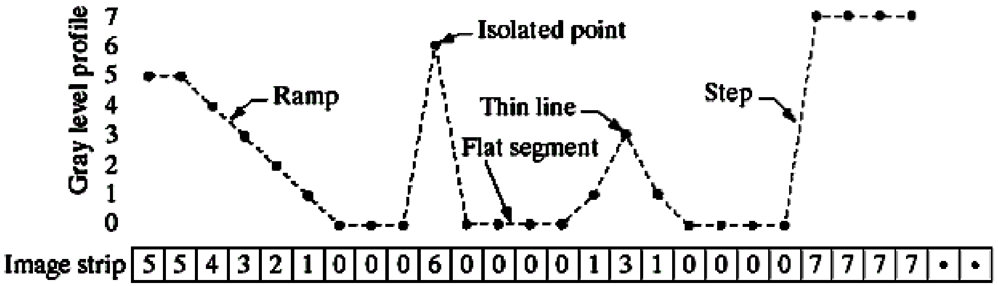
Sharpening filters are based on *spatial differentiation*

Spatial Differentiation

Differentiation measures the *rate of change* of a function

Let's consider a simple 1 dimensional example





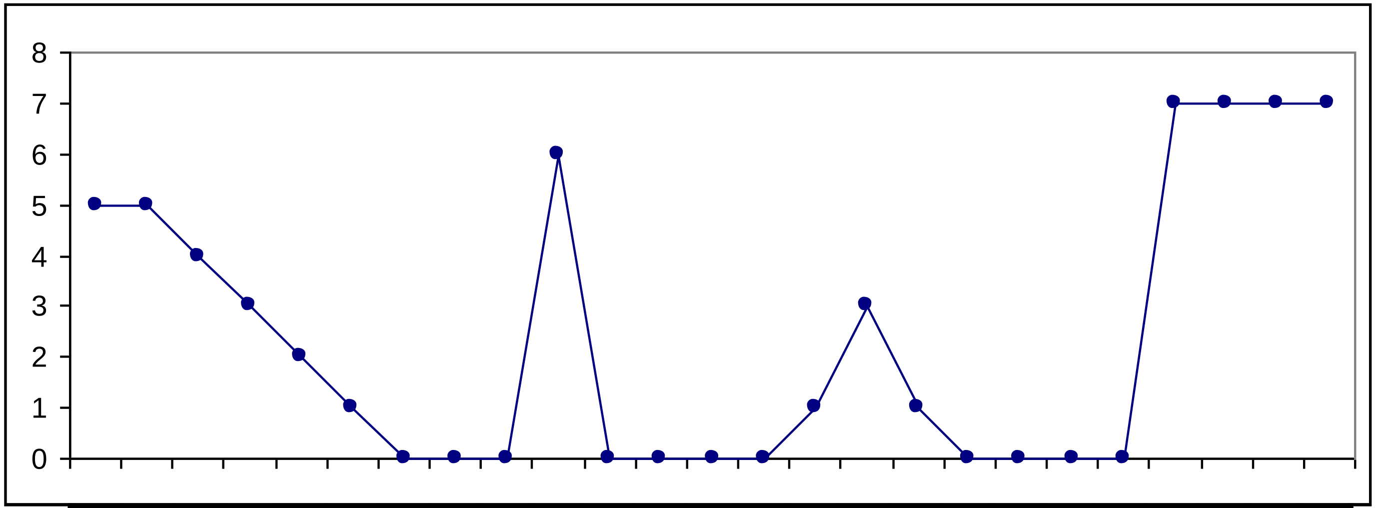
1st Derivative

The formula for the 1st derivative of a function is as follows:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

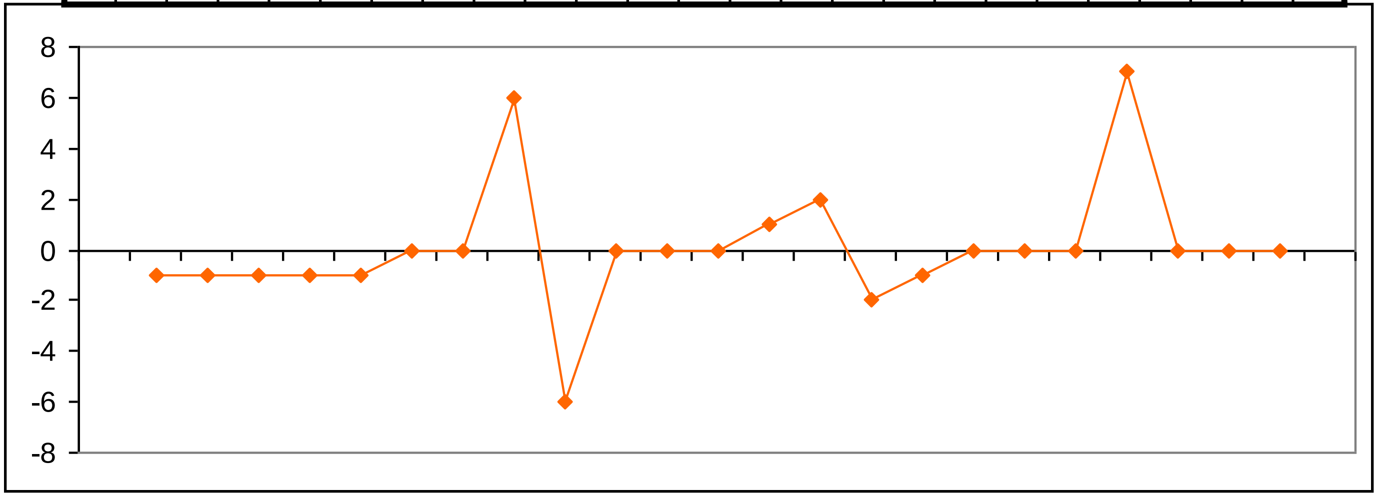
It's just the difference between subsequent values and measures the rate of change of the function

1^{st} Derivative (cont…)



5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

	0	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	



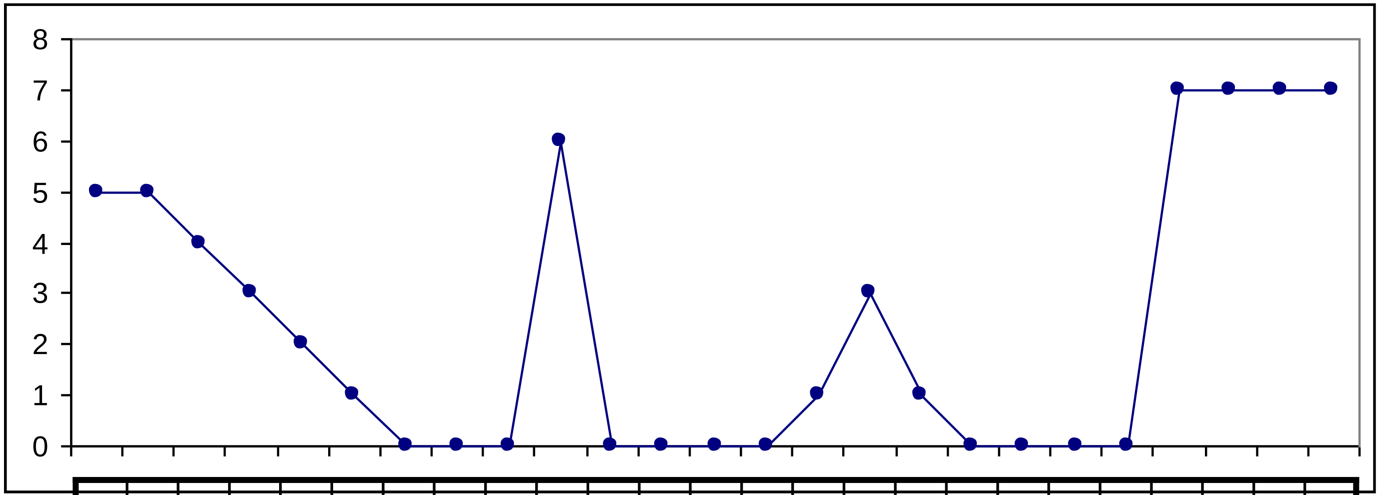
2^{nd} Derivative

The formula for the 2^{nd} derivative of a function is as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

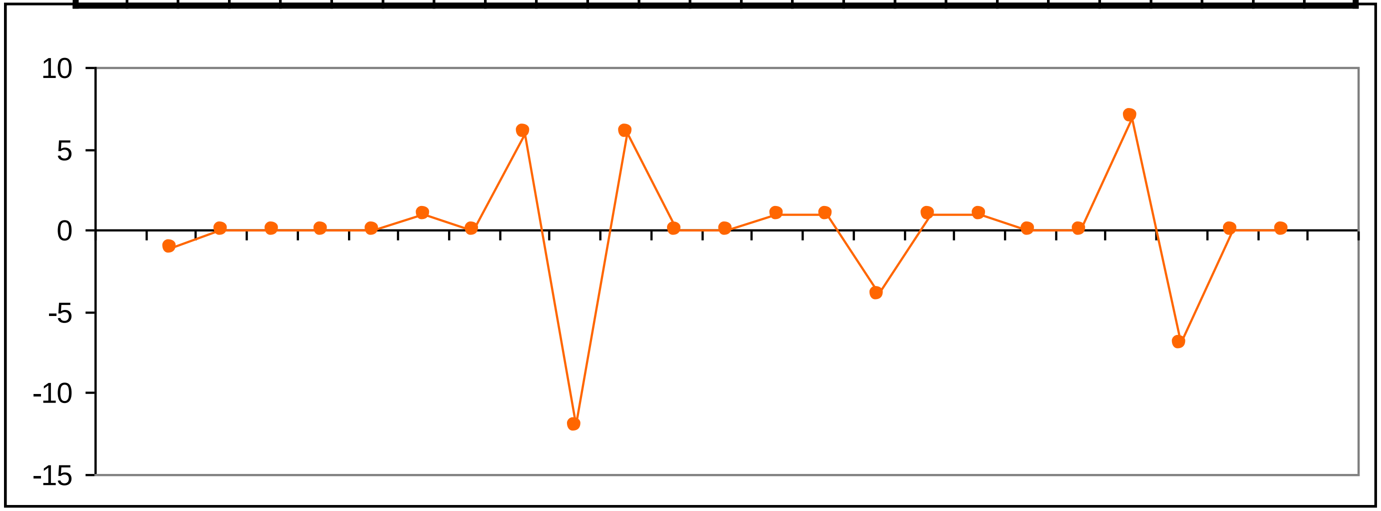
Simply takes into account the values both before and after the current value

2nd Derivative (cont…)



5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



1^{st} & 2^{nd} Derivatives

Comparing the 1^{st} and 2^{nd} derivatives we can conclude the following:

- 1^{st} order derivatives generally produce thicker edges
- 2^{nd} order derivatives have a stronger response to fine detail e.g. thin lines
- 1^{st} order derivatives have stronger response to grey level step
- 2^{nd} order derivatives produce a double response at step changes in grey level

Using Second Derivatives For Image Enhancement

- The 2^{nd} derivative is more useful for image enhancement than the 1^{st} derivative
  - Stronger response to fine detail
  - Simpler implementation
  - We will come back to the 1^{st} order derivative later on
- The first sharpening filter we will look at is the *Laplacian*
  - Isotropic
  - One of the simplest sharpening filters
  - We will look at a digital implementation

The Laplacian

The Laplacian is defined as follows:

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$$

where the partial 1st order derivative in the x direction is defined as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

and in the y direction as follows:

$$\frac{\partial^2 f}{\partial^2 y} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

The Laplacian (cont…)

So, the Laplacian can be given as follows:

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

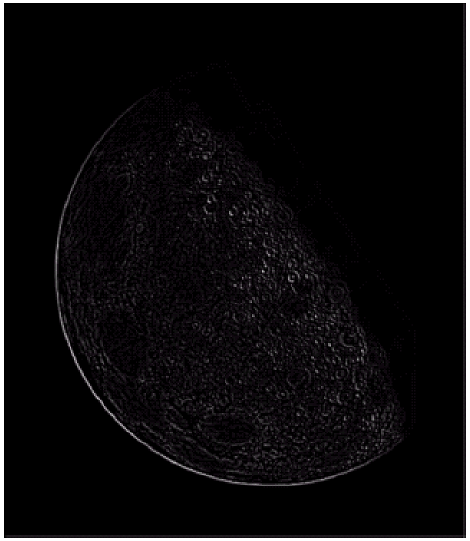
We can easily build a filter based on this

0	1	0
1	-4	1
0	1	0

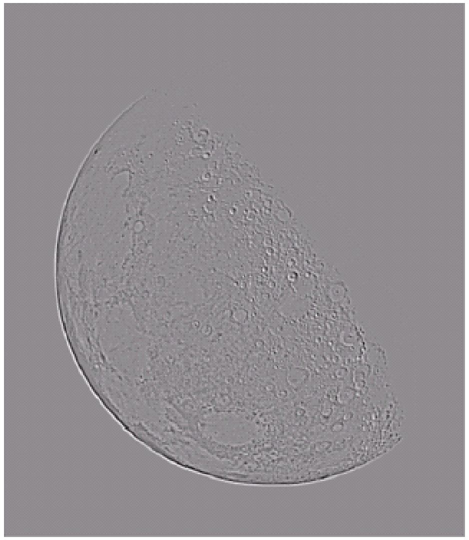
The Laplacian (cont…)

Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities

Original
Image



Laplacian
Filtered Image



Laplacian
Filtered Image
Scaled for Display

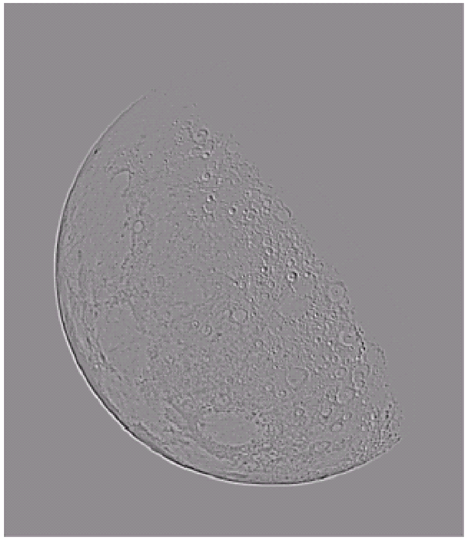
But That Is Not Very Enhanced!

The result of a Laplacian filtering is not an enhanced image

We have to do more work in order to get our final image

Subtract the Laplacian result from the original image to generate our final sharpened enhanced image

$$g(x, y) = f(x, y) - \nabla^2 f$$

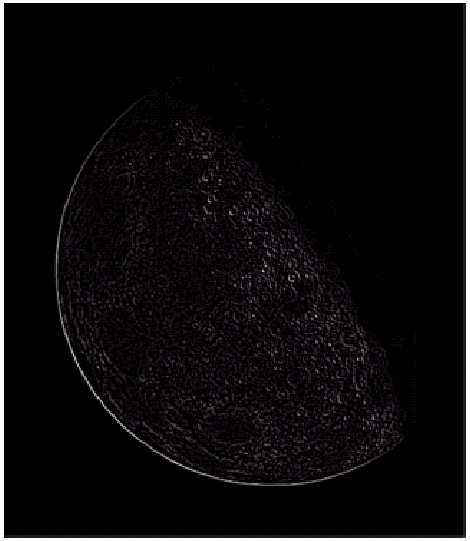


Laplacian Filtered Image Scaled for Display

Laplacian Image Enhancement

Original
Image

-



Laplacian
Filtered Image

=

Sharpened
Image

In the final sharpened image edges and fine detail are much more obvious

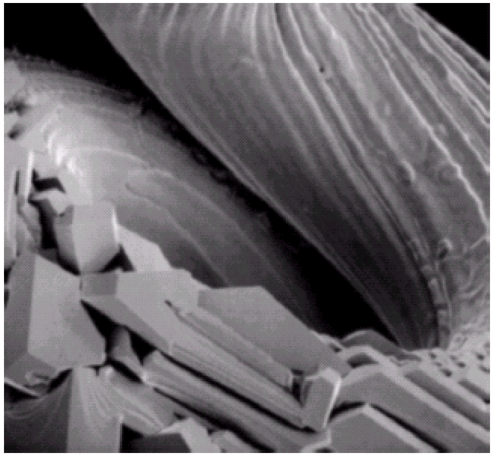
Simplified Image Enhancement

The entire enhancement can be combined into a single filtering operation

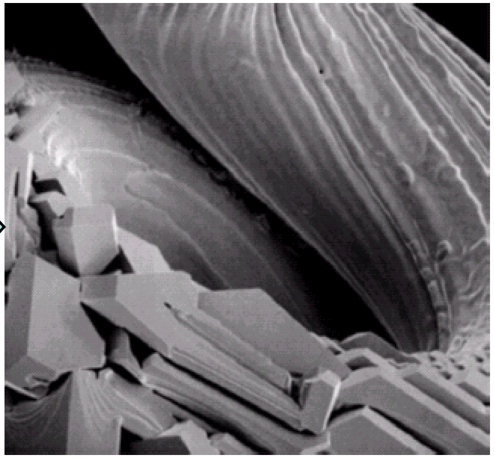
$$
\begin{aligned}
g(x, y) &= f(x, y) - \nabla^2 f \\
&= f(x, y) - [f(x+1, y) + f(x-1, y) \\
&\qquad + f(x, y+1) + f(x, y-1) \\
&\qquad - 4f(x, y)] \\
&= 5f(x, y) - f(x+1, y) - f(x-1, y) \\
&\qquad - f(x, y+1) - f(x, y-1)
\end{aligned}
$$

Simplified Image Enhancement (cont…)

This gives us a new filter which does the whole job for us in one step



0	-1	0
-1	5	-1
0	-1	0



Variants On The Simple Laplacian

There are lots of slightly different versions of the Laplacian that can be used:

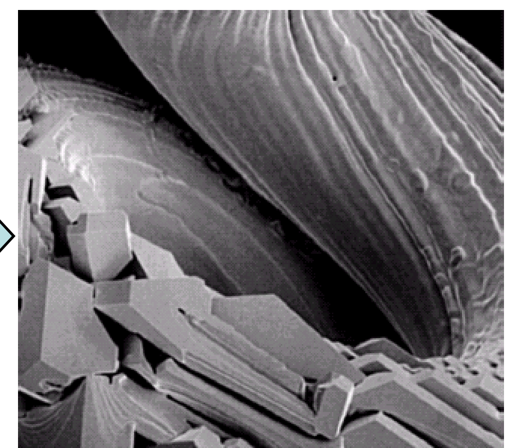
0	1	0
1	-4	1
0	1	0

Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian

-1	-1	-1
-1	9	-1
-1	-1	-1



1^{st} Derivative Filtering

Implementing 1^{st} derivative filters is difficult in practice

For a function $f(x, y)$ the gradient of f at coordinates (x, y) is given as the column vector:

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

1^{st} Derivative Filtering (cont…)

The magnitude of this vector is given by:

$$\nabla f = mag(\nabla \mathrm{f})$$
$$= \left[G_x^2 + G_y^2\right]^{1/2}$$
$$= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2}$$

For practical reasons this can be simplified as:

$$\nabla f \approx \left|G_x\right| + \left|G_y\right|$$

1^{st} Derivative Filtering (cont…)

There is some debate as to how best to calculate these gradients but we will use:

$$\nabla f \approx \left|\left(z_7 + 2z_8 + z_9\right) - \left(z_1 + 2z_2 + z_3\right)\right|$$
$$+ \left|\left(z_3 + 2z_6 + z_9\right) - \left(z_1 + 2z_4 + z_7\right)\right|$$

which is based on these coordinates

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sobel Operators

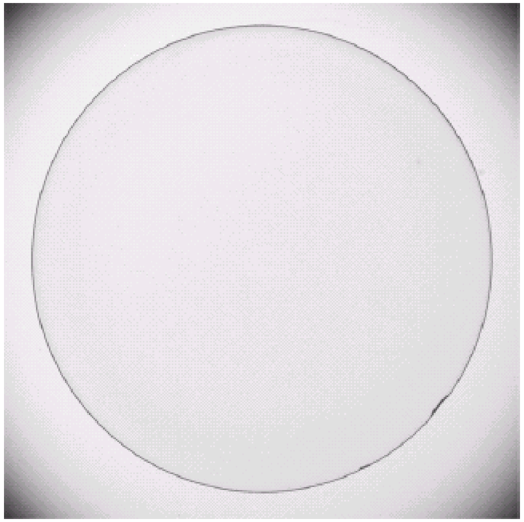
Based on the previous equations we can derive the *Sobel Operators*

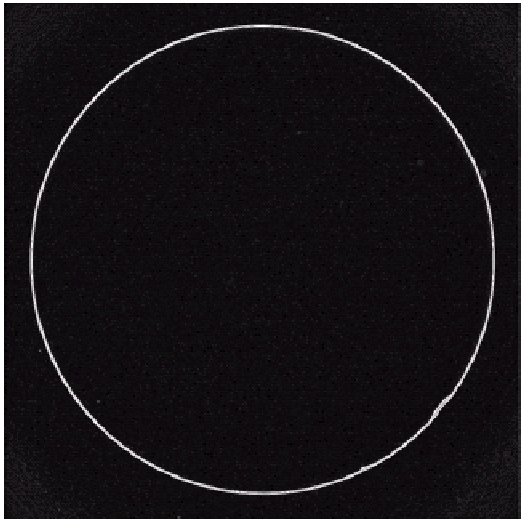
-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

To filter an image it is filtered using both operators the results of which are added together

Sobel Example





An image of a contact lens which is enhanced in order to make defects (at four and five o'clock in the image) more obvious

Sobel filters are typically used for edge detection

Combining Spatial Enhancement Methods

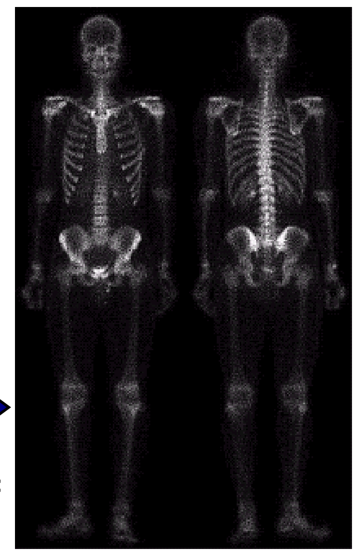
- Successful image enhancement is typically not achieved using a single operation
- Rather we combine a range of techniques in order to achieve a final result
- This example will focus on enhancing the bone scan to the right

Combining Spatial Enhancement Methods (cont…)

(a)

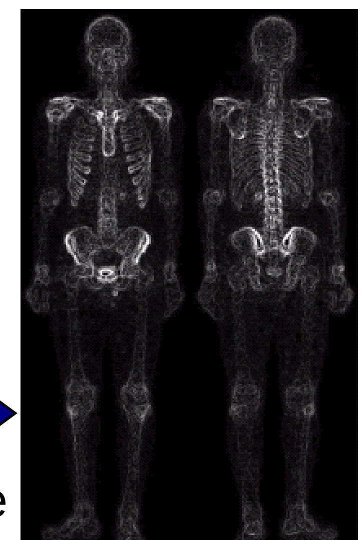
(b)

Laplacian filter of bone scan (a)



(c)

Sharpened version of bone scan achieved by subtracting (a) and (b)



(d)

Sobel filter of bone scan (a)

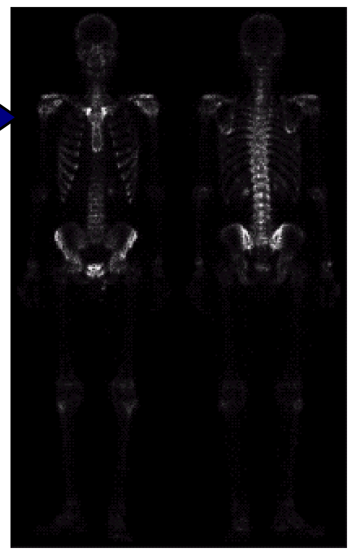
Combining Spatial Enhancement Methods (cont…)

(e)

Image (d) smoothed with a 5*5 averaging filter

The product of (c) and (e) which will be used as a mask

(f)



Sharpened image which is sum of (a) and (f)

(g)

Result of applying a power-law trans. to (g)

(h)

Combining Spatial Enhancement Methods (cont…)

Compare the original and final images

Summary

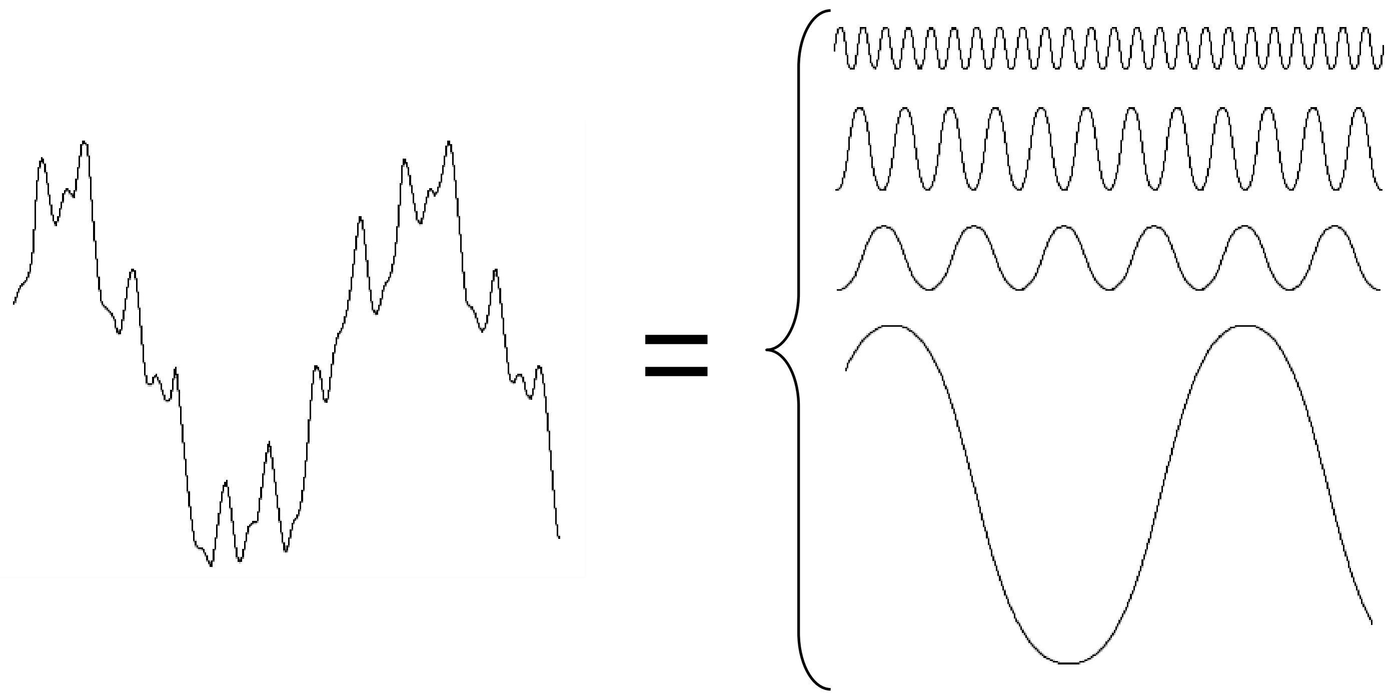
In this lecture we looked at:

- Sharpening filters
  - 1st derivative filters
  - 2nd derivative filters
- Combining filtering techniques

Digital Image Processing

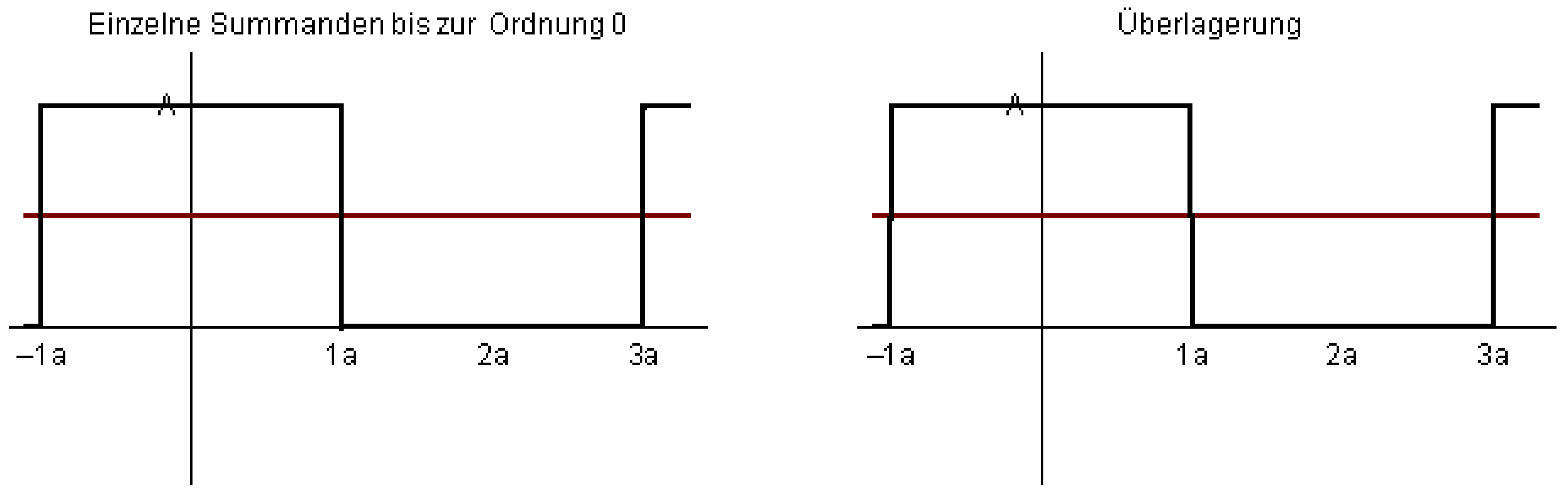
Image Enhancement:
Filtering in the Frequency Domain

The Big Idea



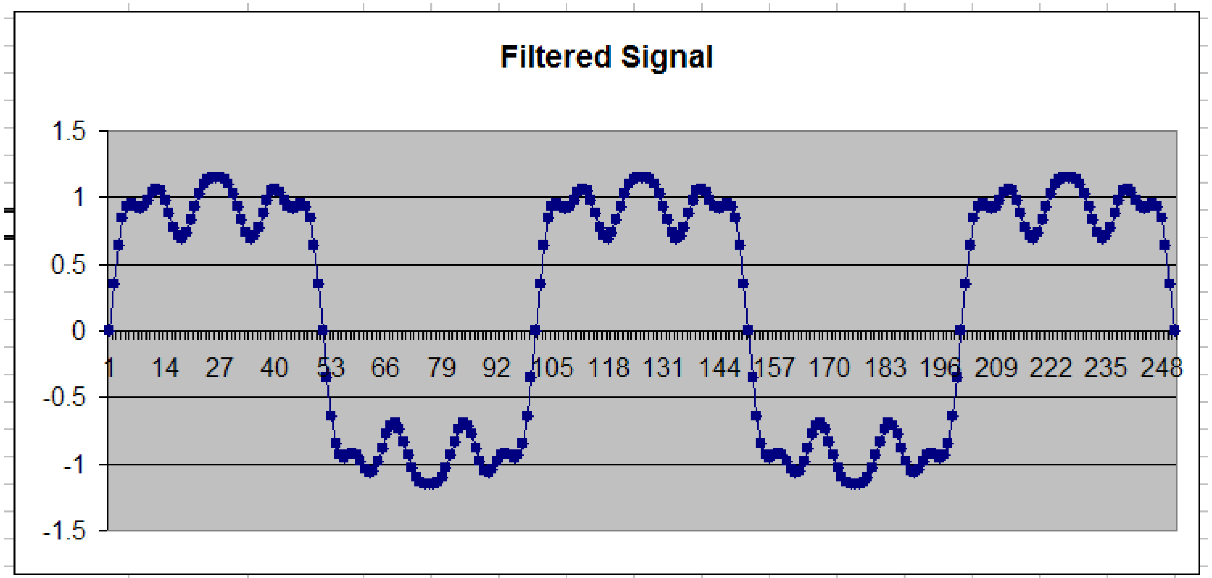
Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient – a *Fourier series*

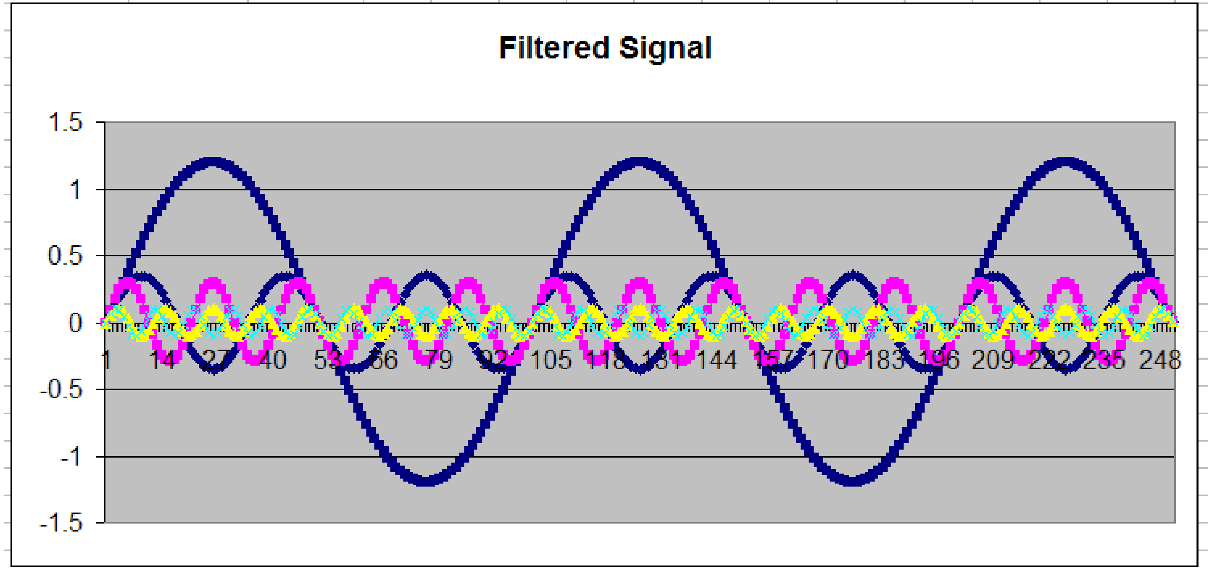
The Big Idea (cont…)



Notice how we get closer and closer to the original function as we add more and more frequencies

The Big Idea (cont…)





Frequency domain signal processing example in Excel

The Discrete Fourier Transform (DFT)

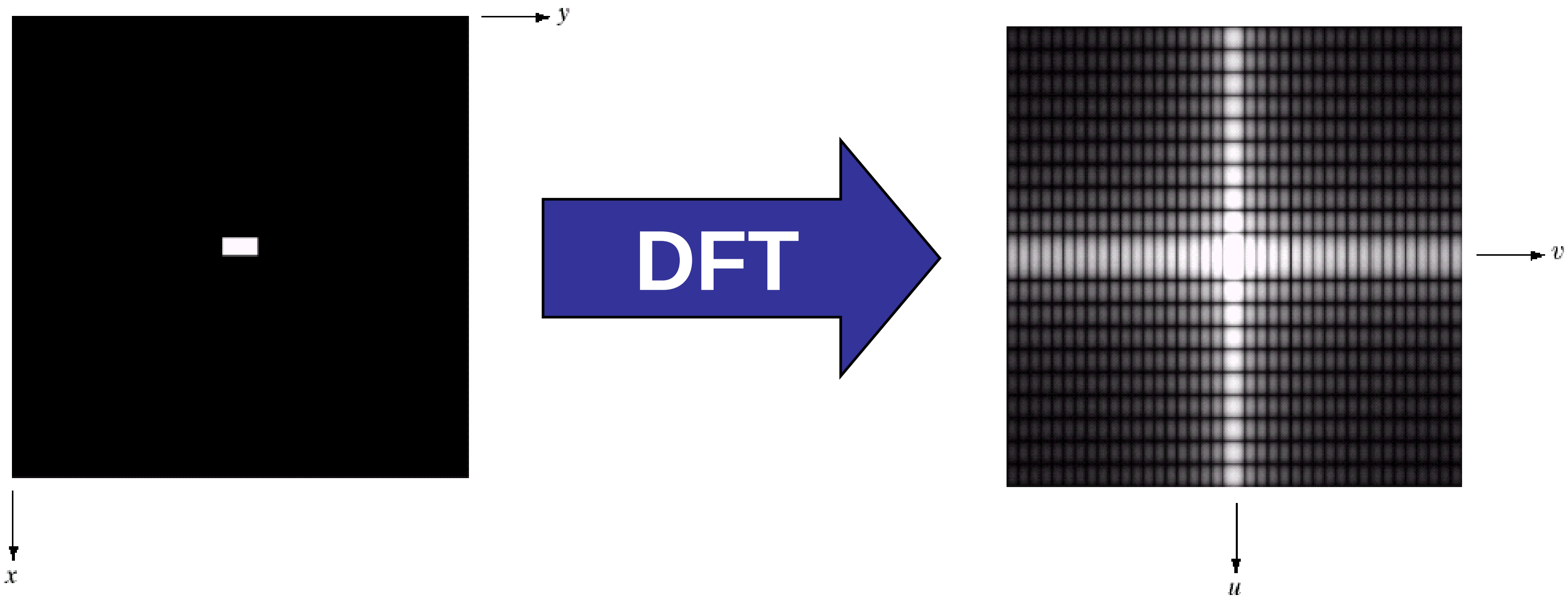
The *Discrete Fourier Transform* of $f(x, y)$, for x = 0, 1, 2…M-1 and y = 0,1,2…N-1, denoted by $F(u, v)$, is given by the equation:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

for u = 0, 1, 2…M-1 and v = 0, 1, 2…N-1.

DFT & Images

The DFT of a two dimensional image can be visualised by showing the spectrum of the images component frequencies



The Inverse DFT

It is really important to note that the Fourier transform is completely **reversible**

The inverse DFT is given by:

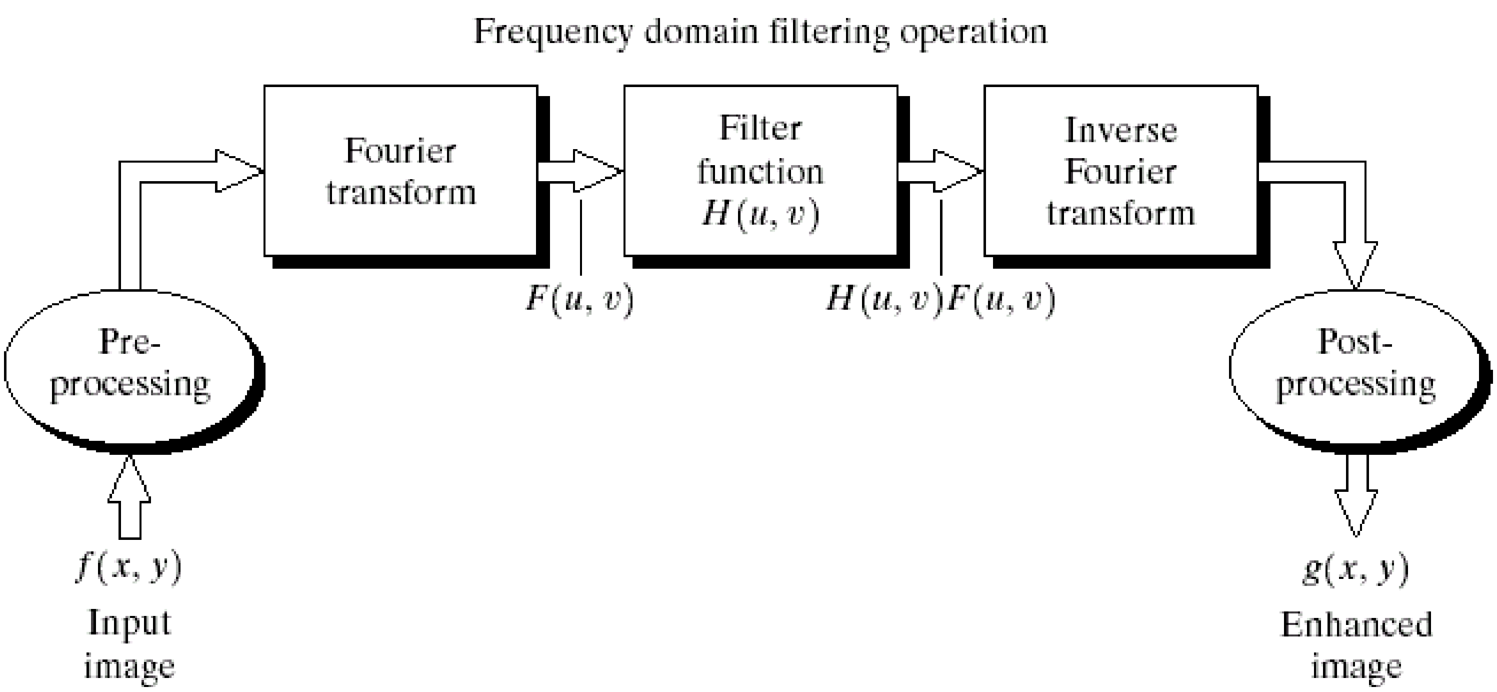
$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{j2\pi(ux/M+vy/N)}$$

for x = 0, 1, 2…M-1 and y = 0, 1, 2…N-1

The DFT and Image Processing

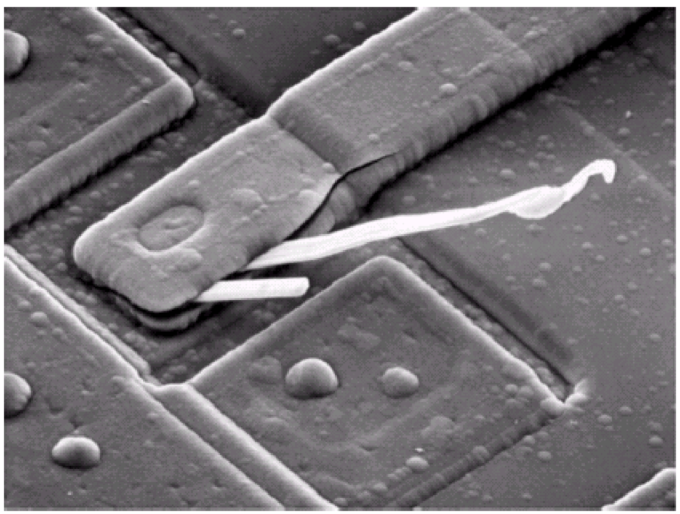
To filter an image in the frequency domain:

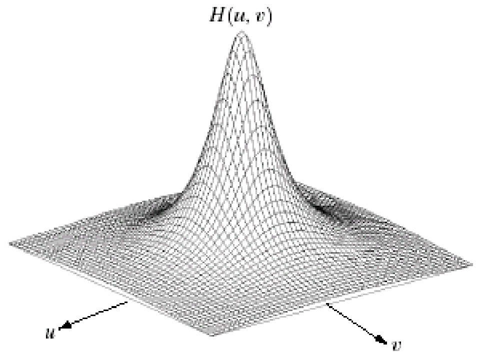
1. Compute $F(u,v)$ the DFT of the image
2. Multiply $F(u,v)$ by a filter function $H(u,v)$
3. Compute the inverse DFT of the result

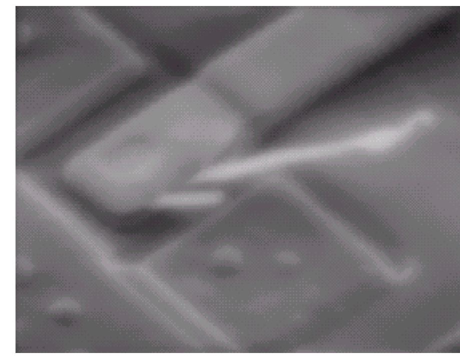


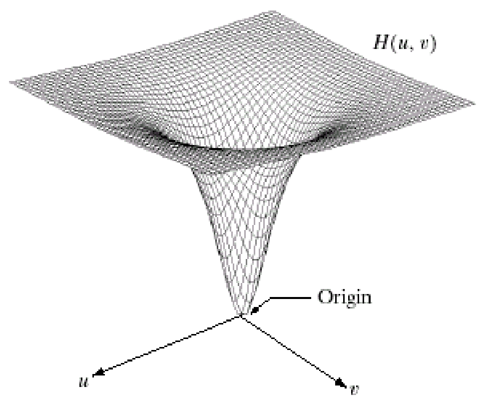
Some Basic Frequency Domain Filters

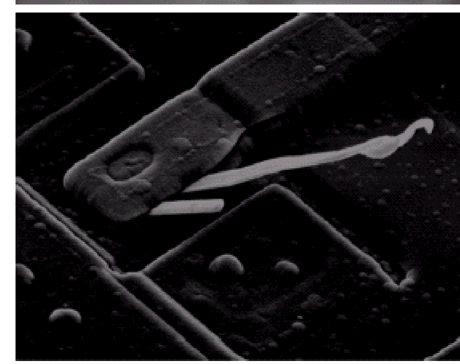
Low Pass Filter











High Pass Filter

Smoothing Frequency Domain Filters

Smoothing is achieved in the frequency domain by dropping out the high frequency components (*Low pass filters*)

The basic model for filtering is:

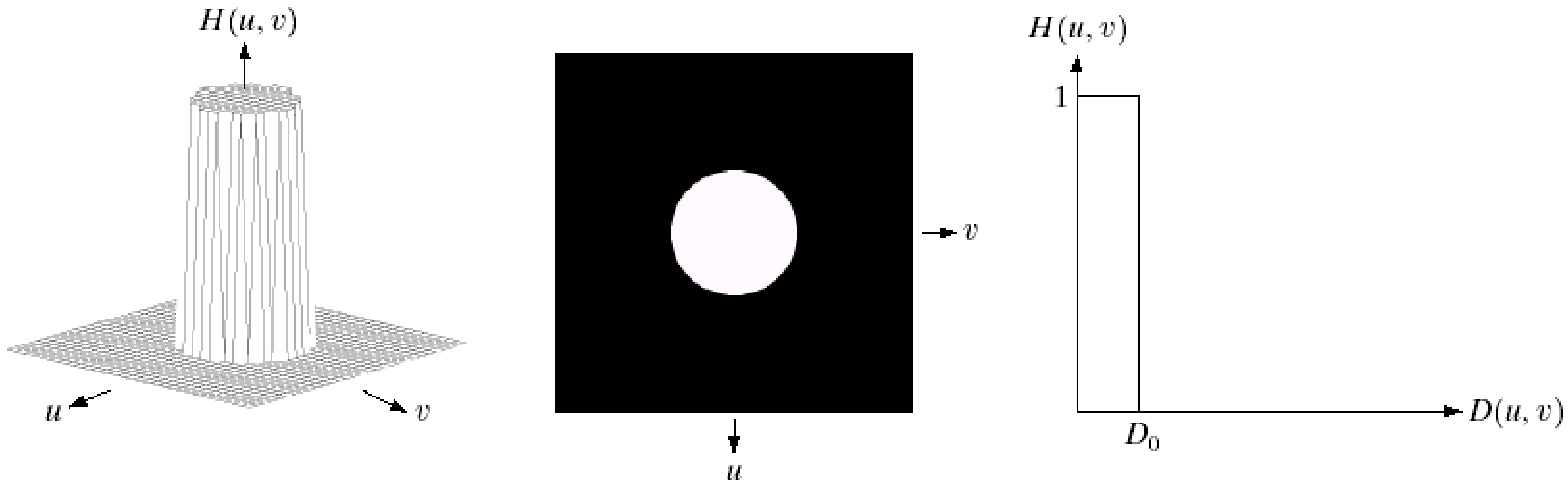
$$G(u,v) = H(u,v)F(u,v)$$

where $F(u,v)$ is the Fourier transform of the image being filtered and $H(u,v)$ is the filter transform function

Low pass filters – only pass the low frequencies, drop the high ones

Ideal Low Pass Filter

Simply cut off all high frequency components that are a specified distance D_0 from the origin of the transform



changing the distance changes the behaviour of the filter

Ideal Low Pass Filter (cont…)

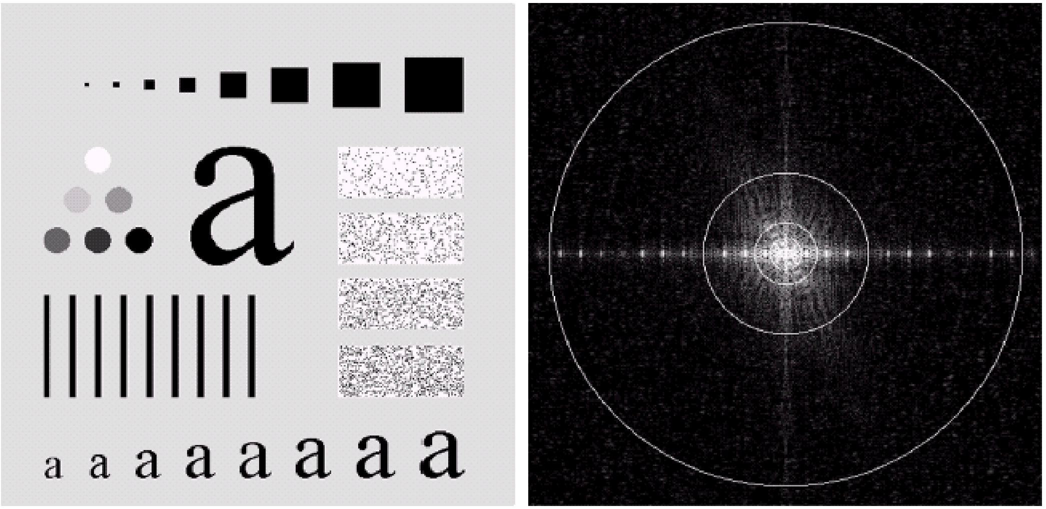
The transfer function for the ideal low pass filter can be given as:

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

where $D(u,v)$ is given as:

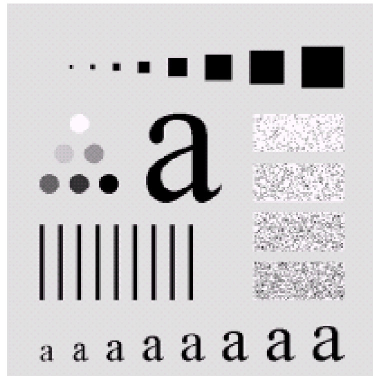
$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

Ideal Low Pass Filter (cont…)

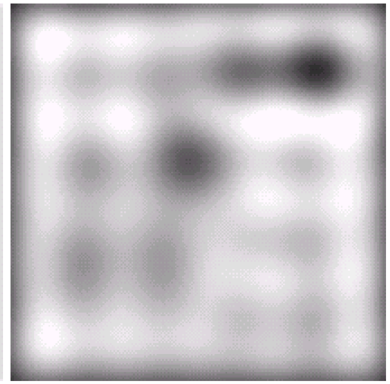


Above we show an image, it's Fourier spectrum and a series of ideal low pass filters of radius 5, 15, 30, 80 and 230 superimposed on top of it

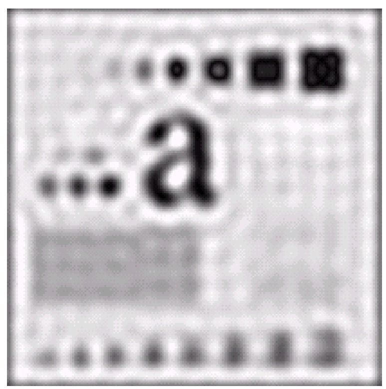
Ideal Low Pass Filter (cont…)



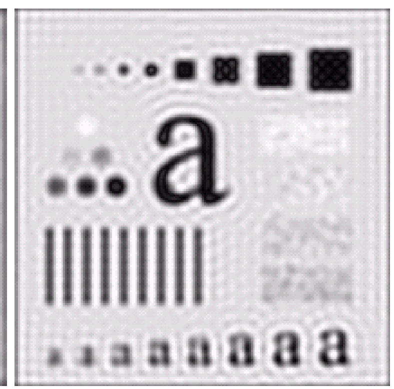
Original image



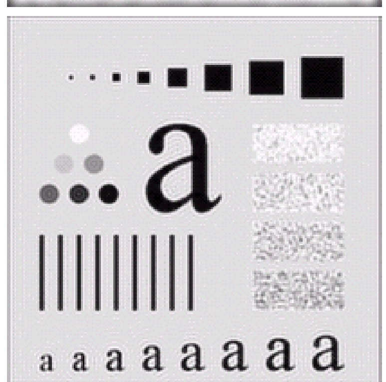
Result of filtering with ideal low pass filter of radius 5



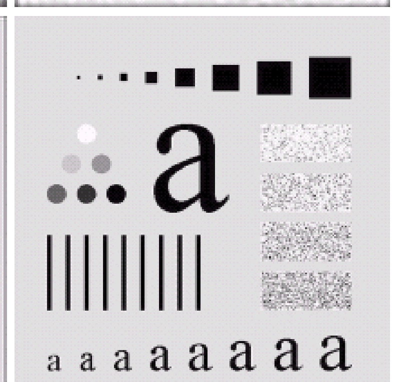
Result of filtering with ideal low pass filter of radius 15



Result of filtering with ideal low pass filter of radius 30



Result of filtering with ideal low pass filter of radius 80



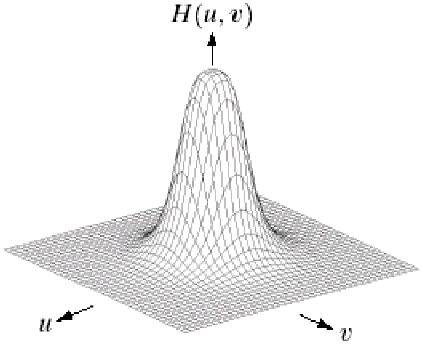
Result of filtering with ideal low pass filter of radius 230

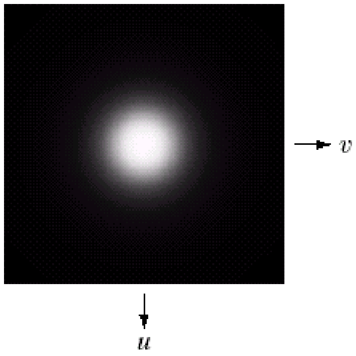
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

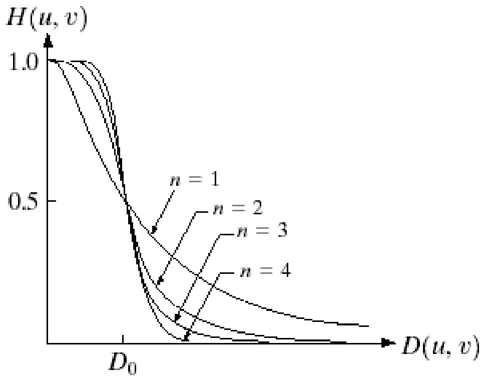
Butterworth Lowpass Filters

The transfer function of a Butterworth lowpass filter of order n with cutoff frequency at distance D_0 from the origin is defined as:

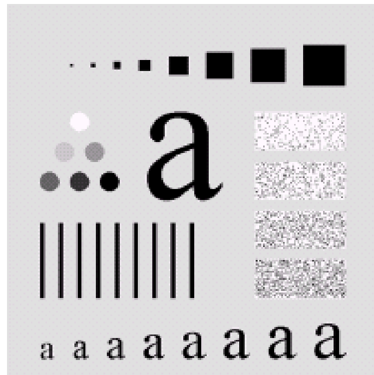
$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$



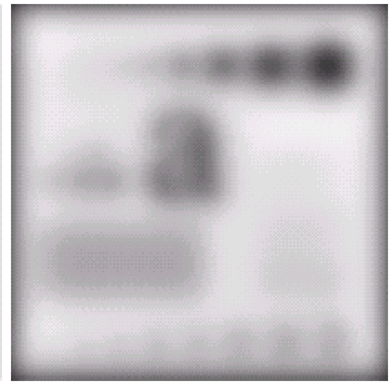




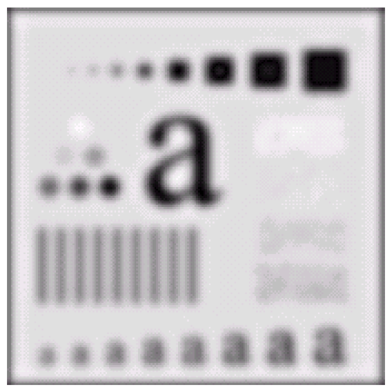
Butterworth Lowpass Filter (cont…)



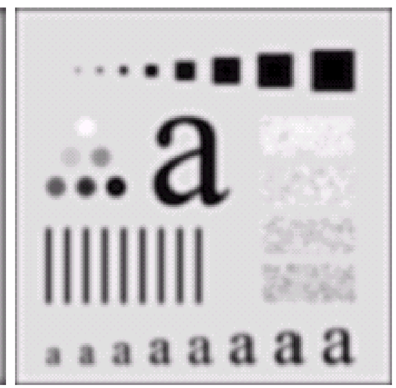
Original image



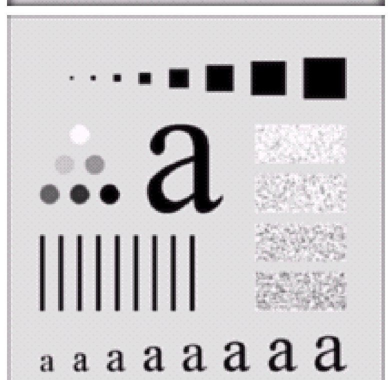
Result of filtering with Butterworth filter of order 2 and cutoff radius 5



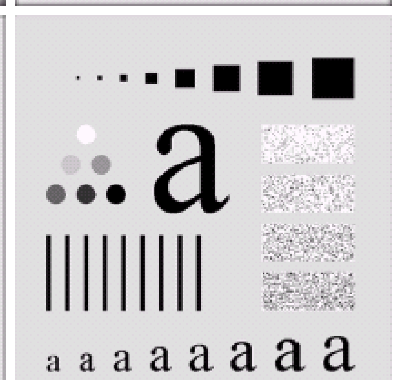
Result of filtering with Butterworth filter of order 2 and cutoff radius 15



Result of filtering with Butterworth filter of order 2 and cutoff radius 30



Result of filtering with Butterworth filter of order 2 and cutoff radius 80

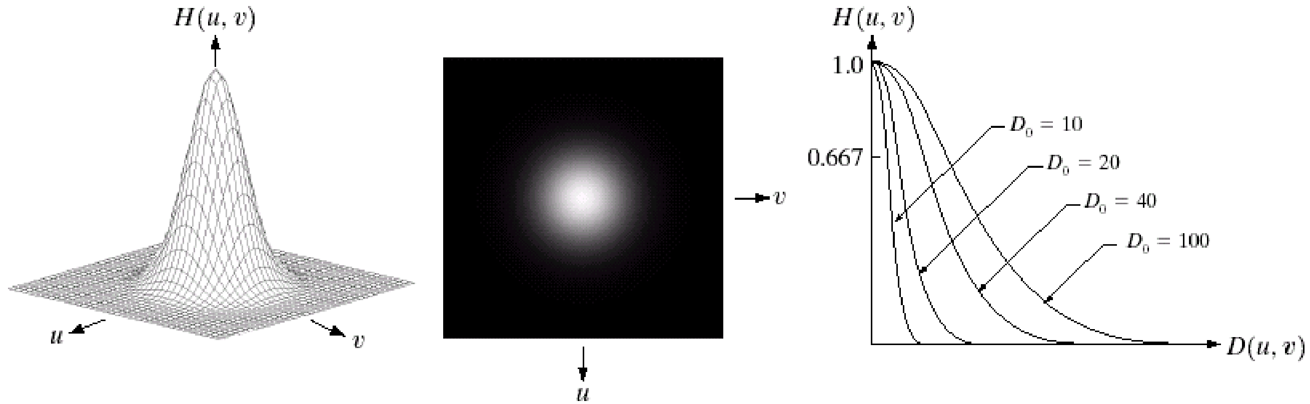


Result of filtering with Butterworth filter of order 2 and cutoff radius 230

Gaussian Lowpass Filters

The transfer function of a Gaussian lowpass filter is defined as:

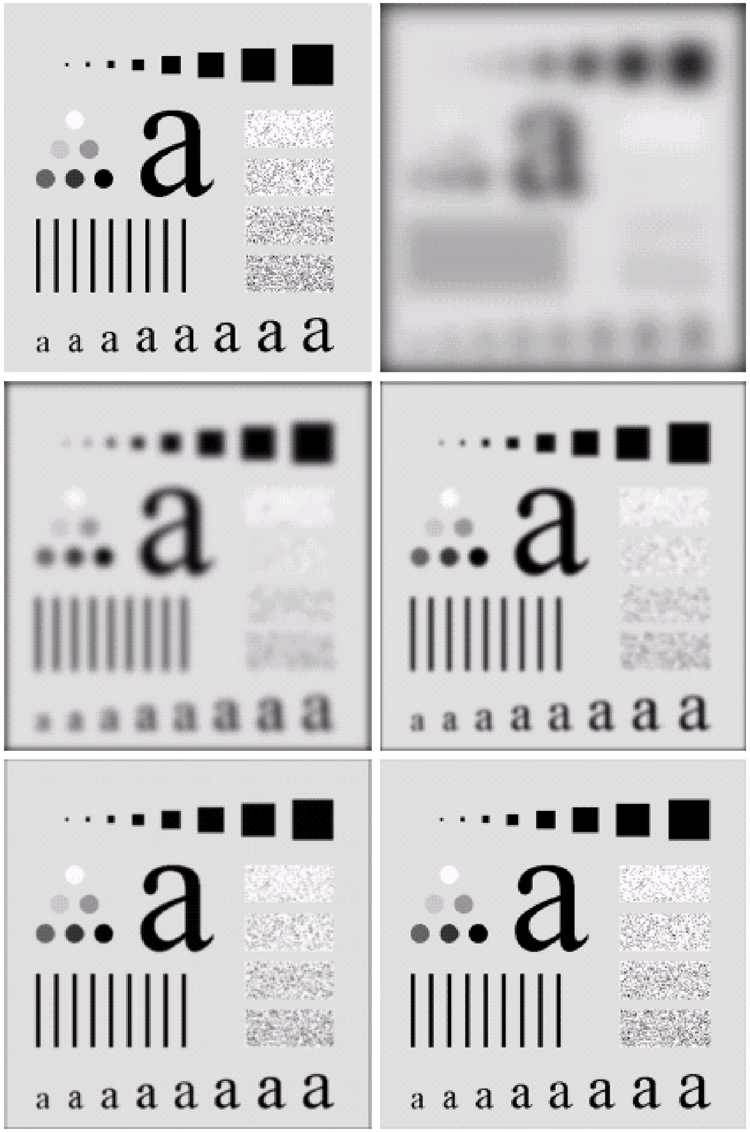
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$



Gaussian Lowpass Filters (cont…)

Original image

Result of filtering with Gaussian filter with cutoff radius 5



Result of filtering with Gaussian filter with cutoff radius 15

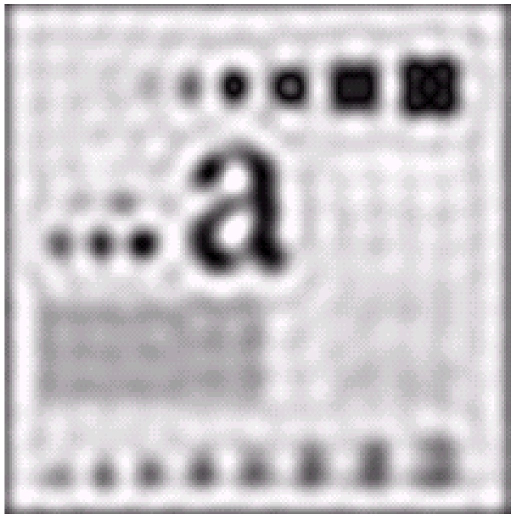
Result of filtering with Gaussian filter with cutoff radius 30

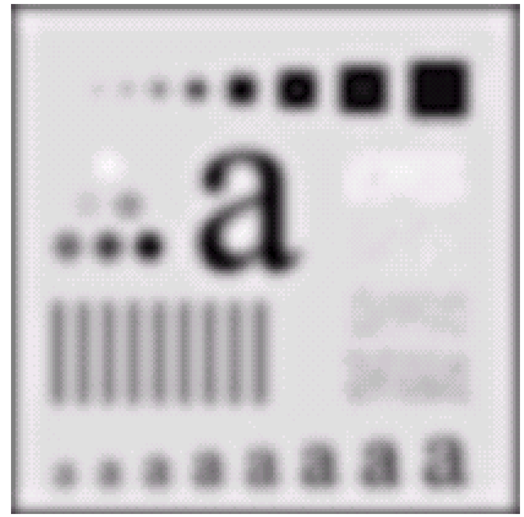
Result of filtering with Gaussian filter with cutoff radius 85

Result of filtering with Gaussian filter with cutoff radius 230

Lowpass Filters Compared

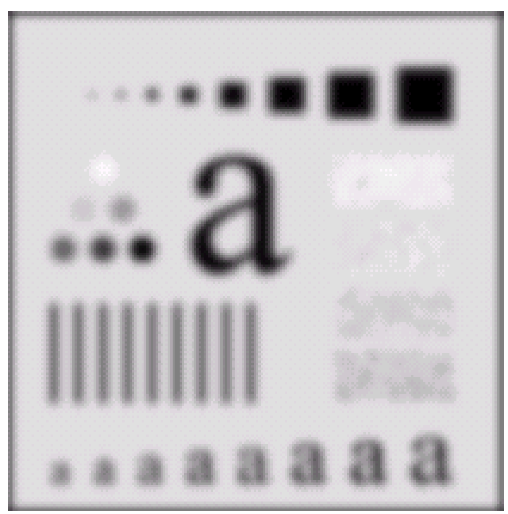
Result of filtering with ideal low pass filter of radius 15





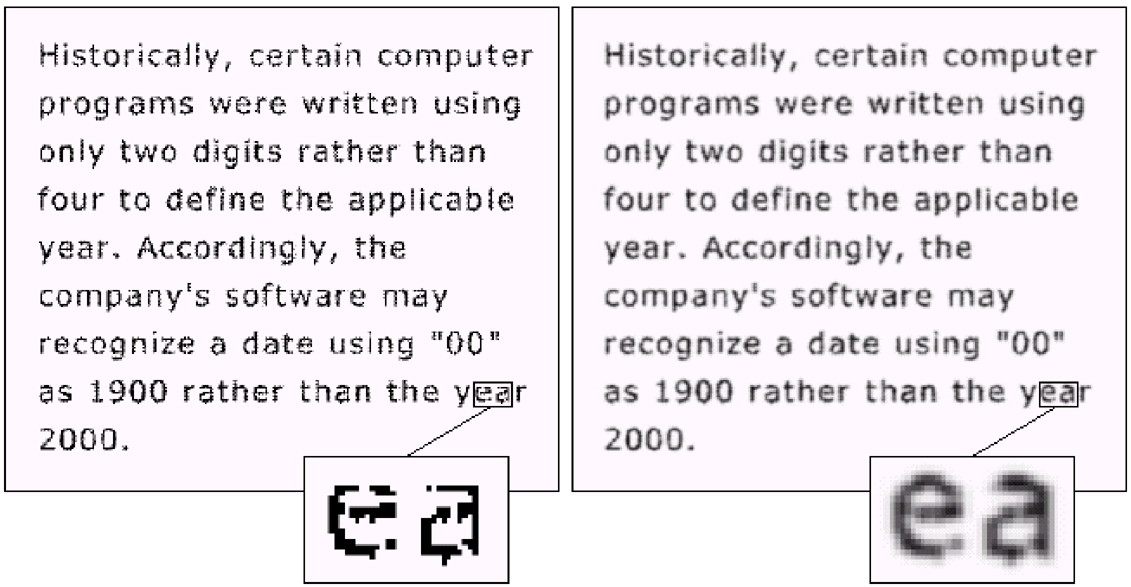
Result of filtering with Butterworth filter of order 2 and cutoff radius 15

Result of filtering with Gaussian filter with cutoff radius 15



Lowpass Filtering Examples

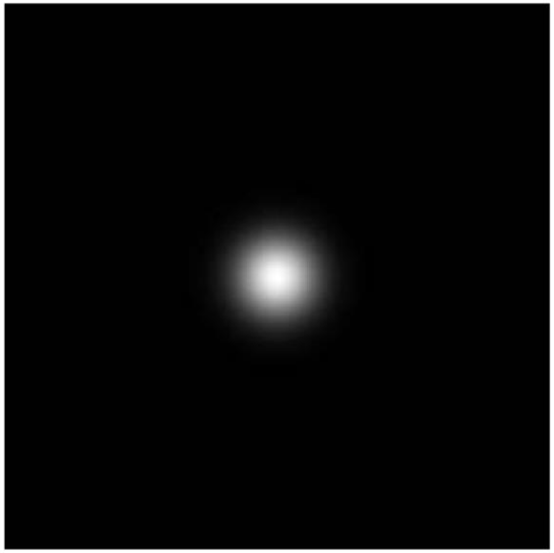
A low pass Gaussian filter is used to connect broken text


Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

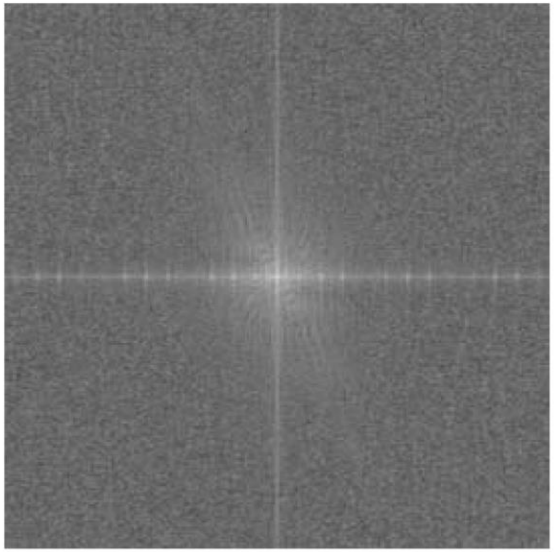
Lowpass Filtering Examples (cont…)

Original image



Gaussian lowpass filter

Spectrum of original image



Processed image

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Sharpening in the Frequency Domain

Edges and fine detail in images are associated with high frequency components

High pass filters – only pass the high frequencies, drop the low ones

High pass frequencies are precisely the reverse of low pass filters, so:

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

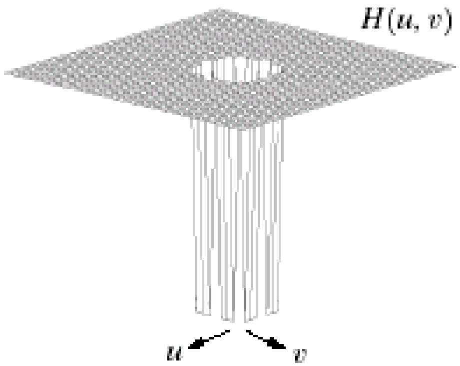
hp:- high path. lp:- low path.

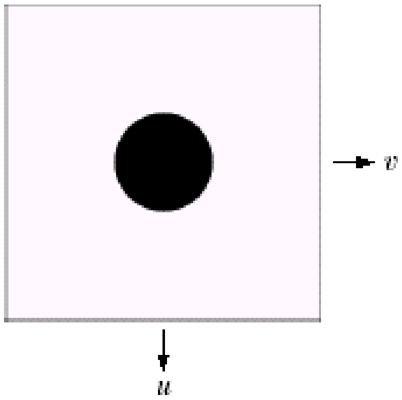
Ideal High Pass Filters

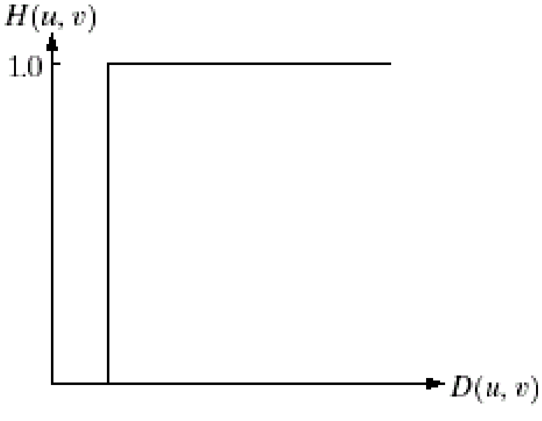
The ideal high pass filter is given as:

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

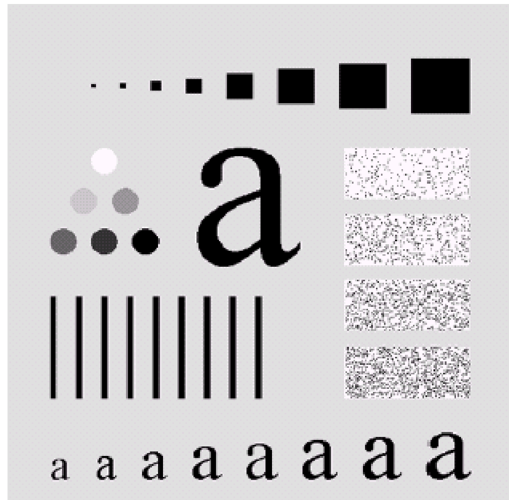
where D_0 is the cut off distance as before

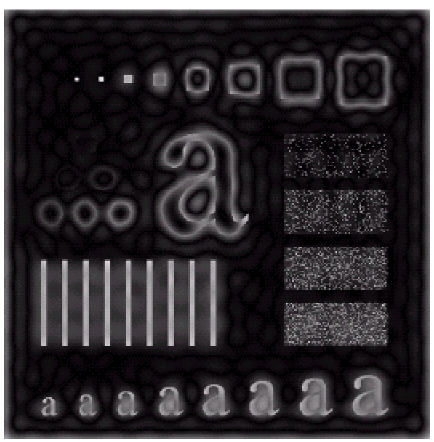




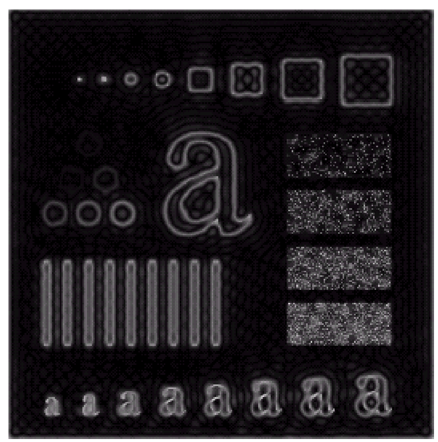


Ideal High Pass Filters (cont…)

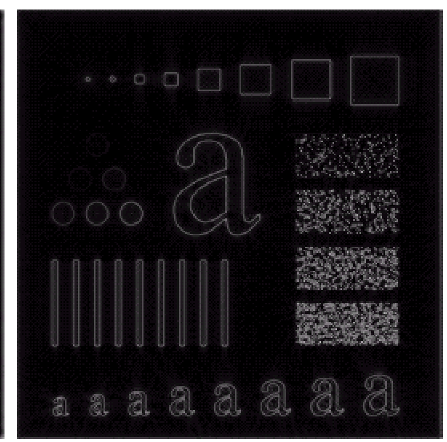




Results of ideal high pass filtering with D_0 = 15



Results of ideal high pass filtering with D_0 = 30



Results of ideal high pass filtering with D_0 = 80

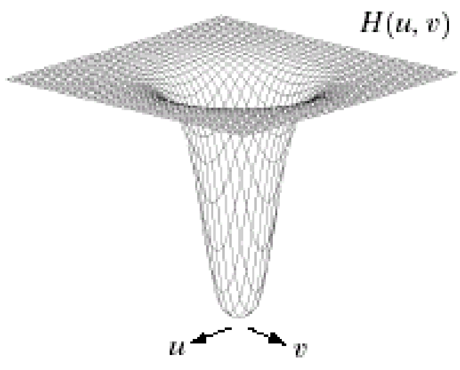
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

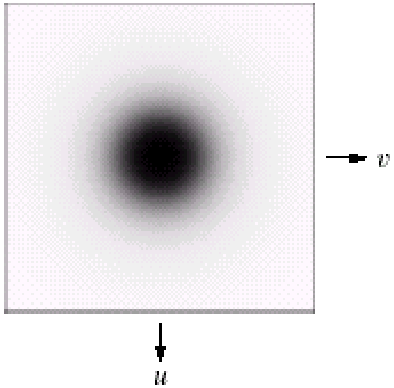
Butterworth High Pass Filters

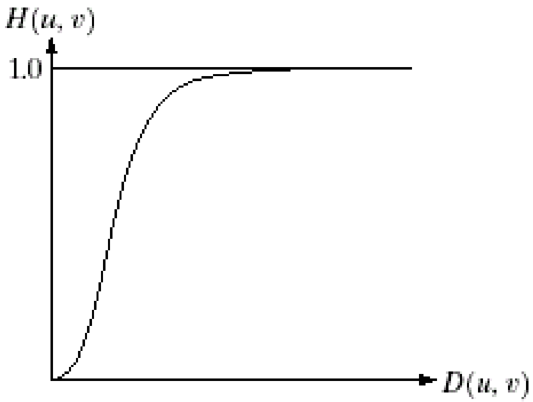
The Butterworth high pass filter is given as:

$$H(u,v) = \frac{1}{1 + [D_0 / D(u,v)]^{2n}}$$

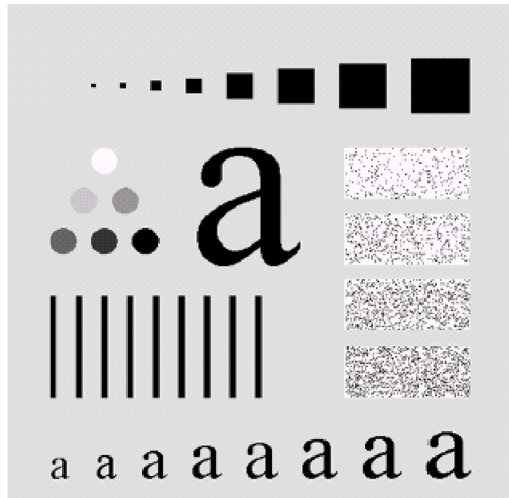
where n is the order and D_0 is the cut off distance as before

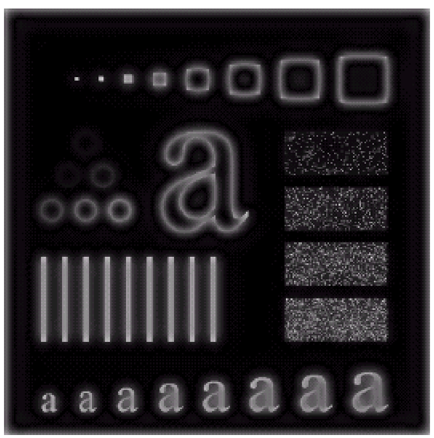




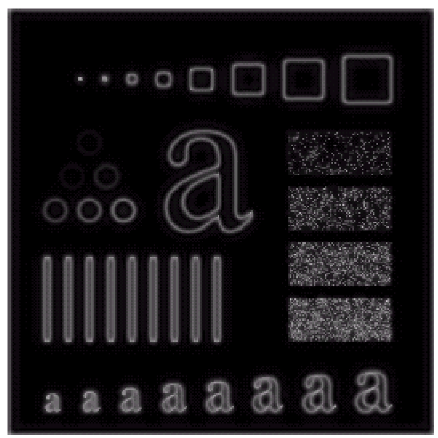


Butterworth High Pass Filters (cont…)

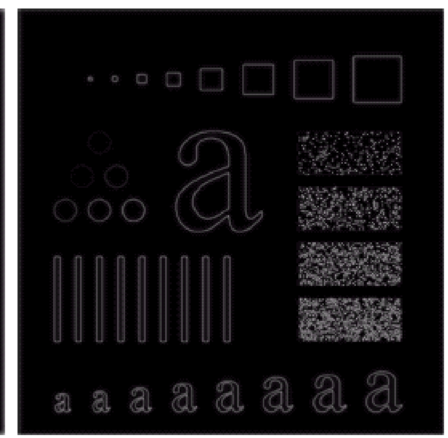




Results of Butterworth high pass filtering of order 2 with $D_0 = 15$



Results of Butterworth high pass filtering of order 2 with $D_0 = 30$



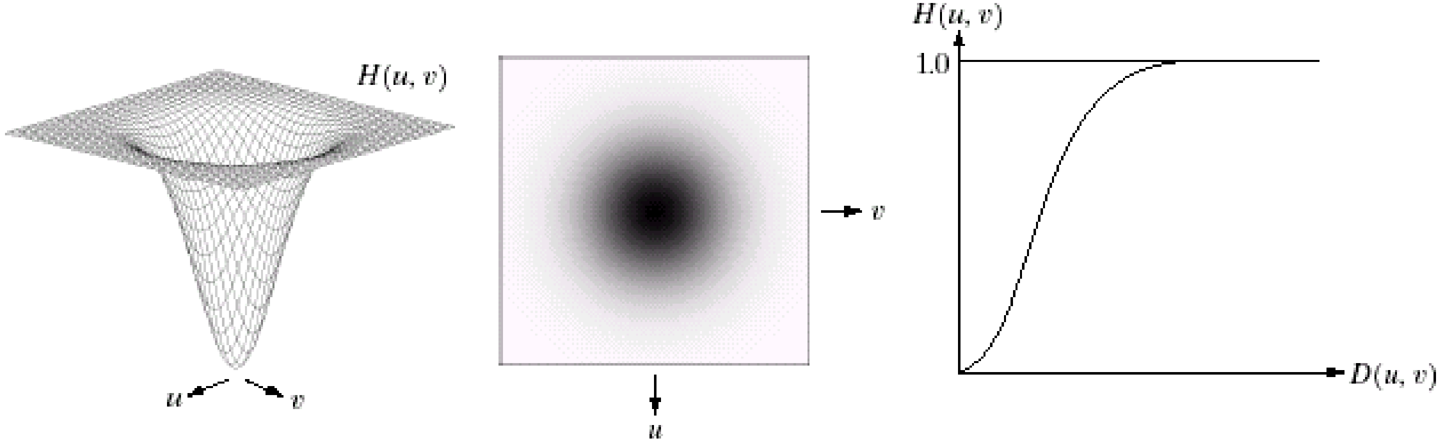
Results of Butterworth high pass filtering of order 2 with $D_0 = 80$

Gaussian High Pass Filters

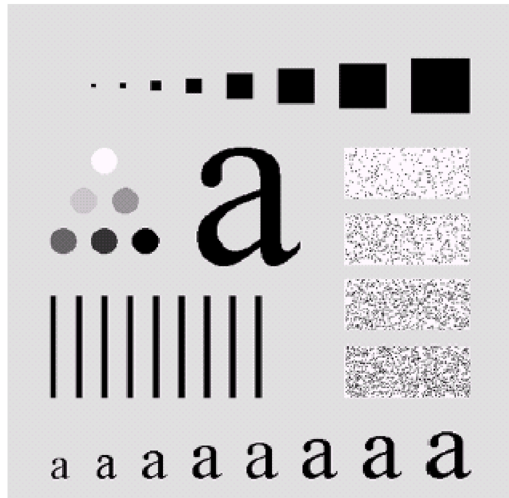
The Gaussian high pass filter is given as:

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

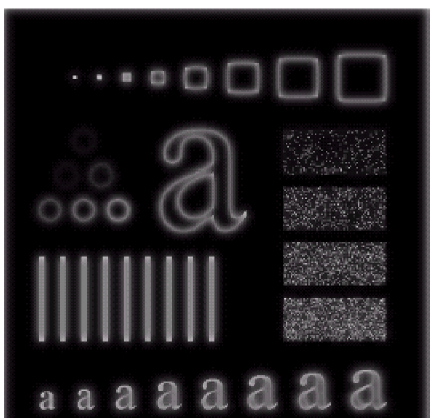
where D_0 is the cut off distance as before

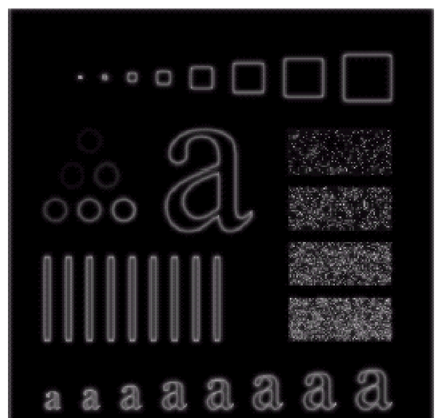


Gaussian High Pass Filters (cont…)



Results of Gaussian high pass filtering with D_0 = 15

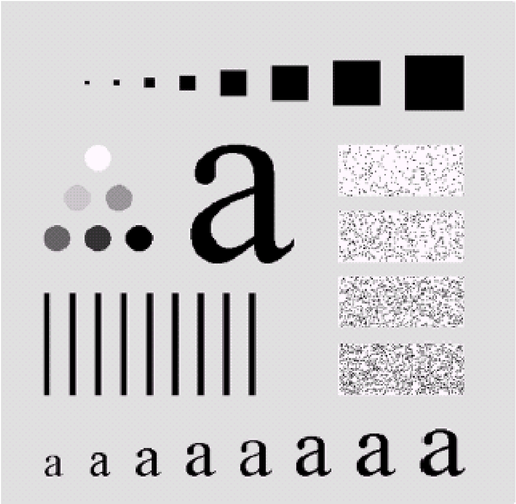


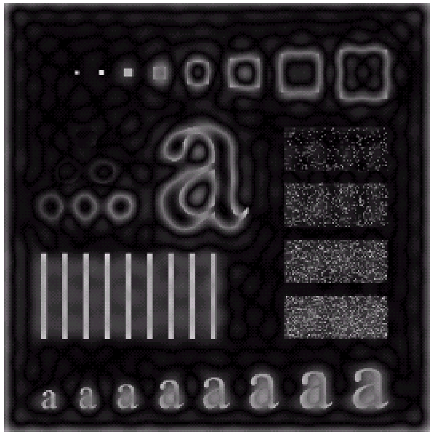


Results of Gaussian high pass filtering with D_0 = 80

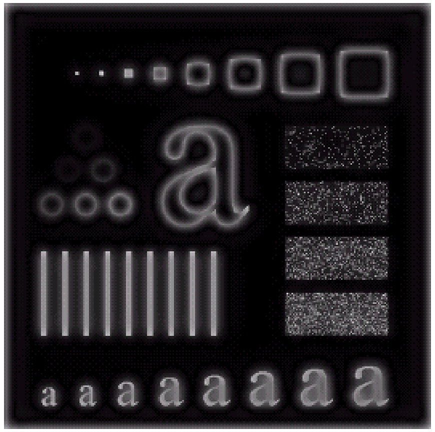
Results of Gaussian high pass filtering with D_0 = 30

Highpass Filter Comparison

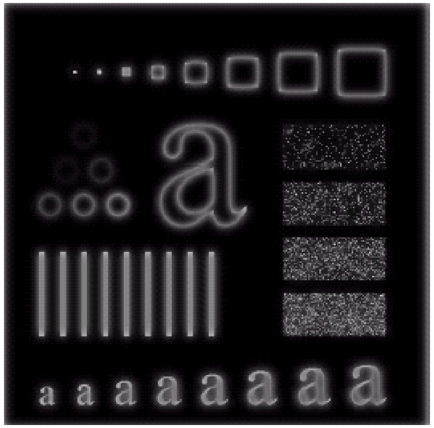




Results of ideal high pass filtering with D_0 = 15

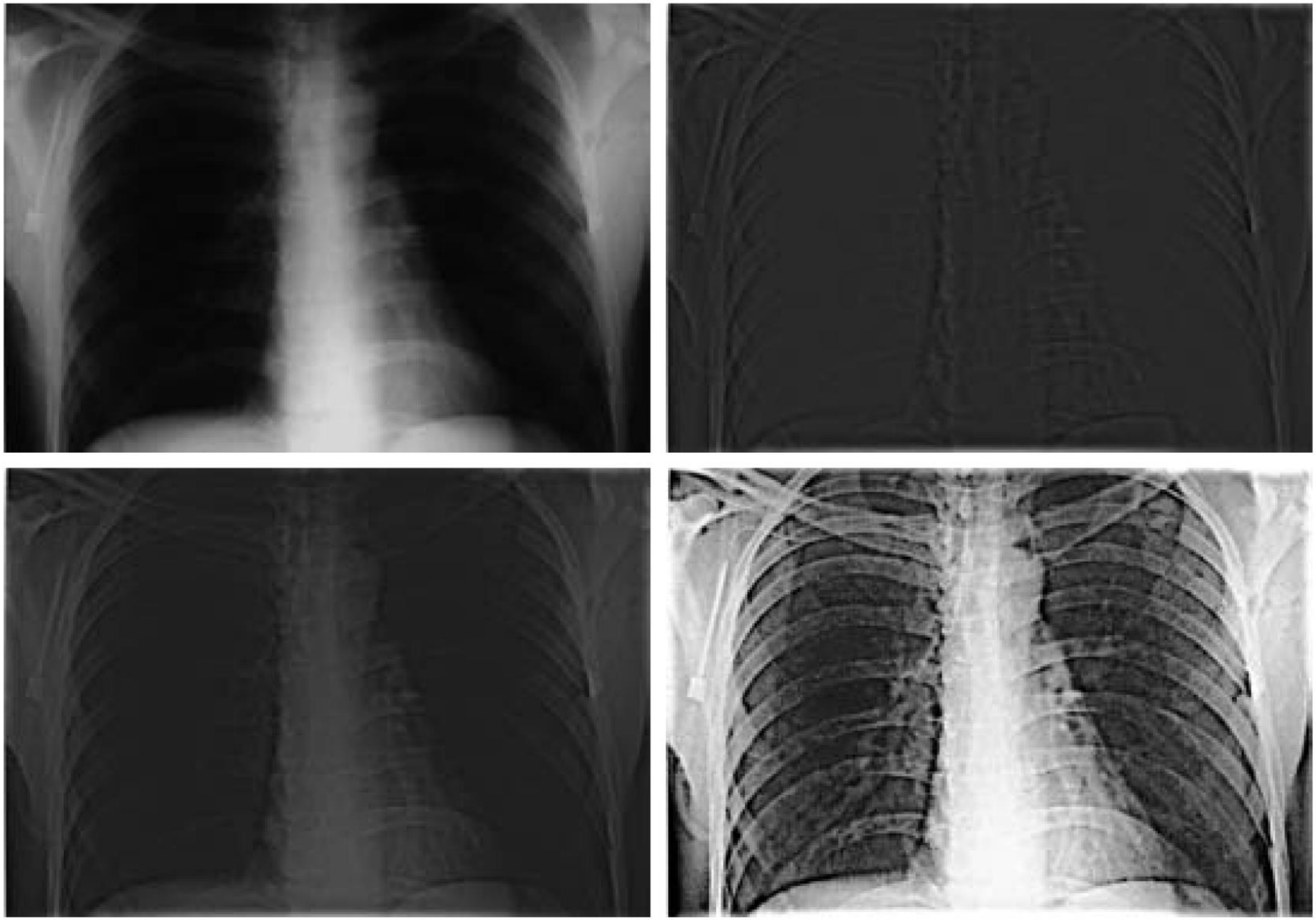


Results of Butterworth high pass filtering of order 2 with D_0 = 15



Results of Gaussian high pass filtering with D_0 = 15

Highpass Filtering Example



Original image

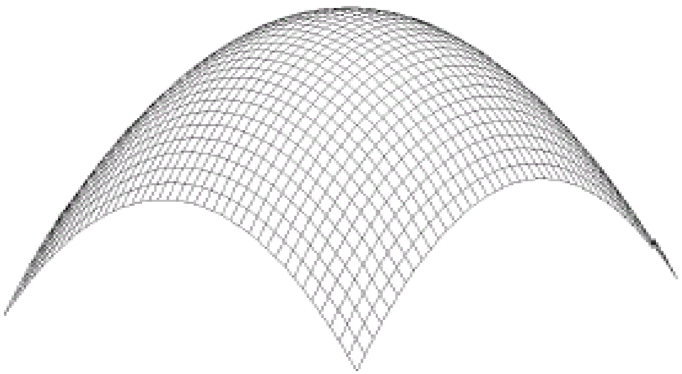
Highpass filtering result

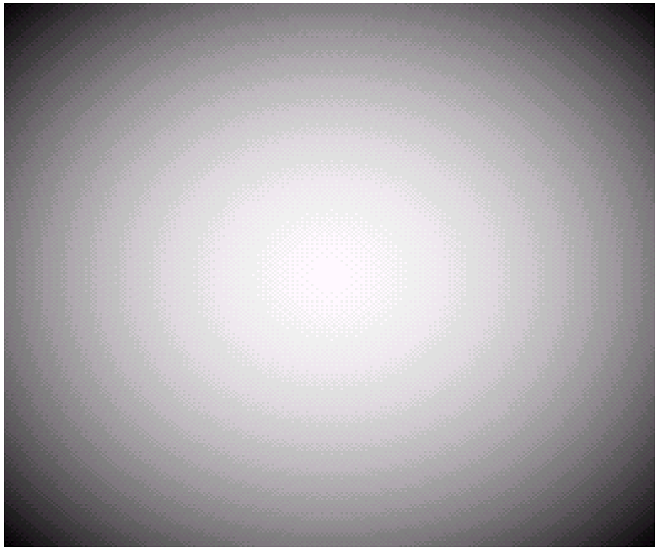
High frequency emphasis result

After histogram equalisation

Laplacian In The Frequency Domain

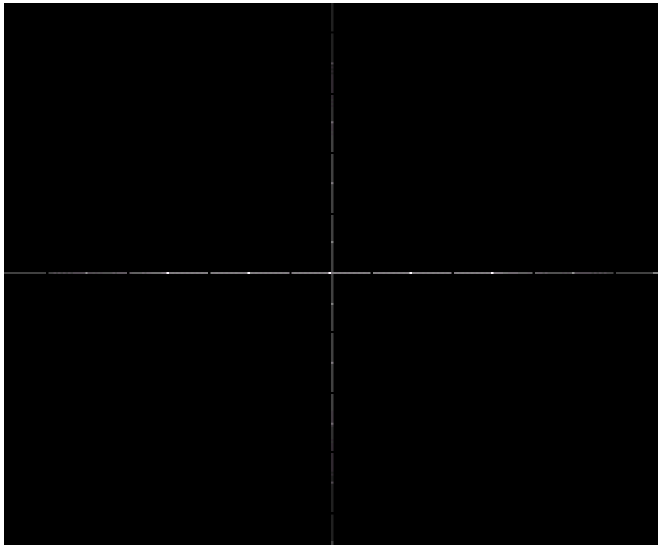
Laplacian in the frequency domain

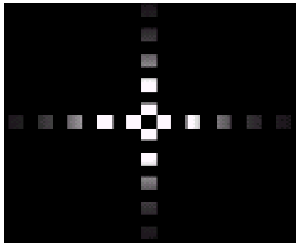


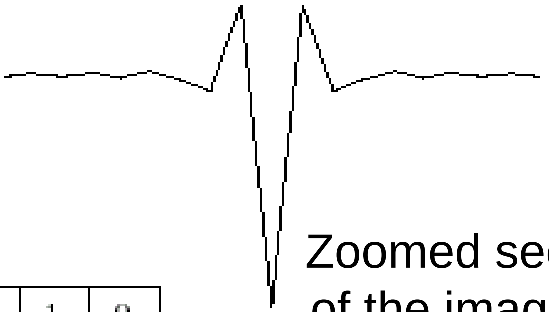


2-D image of Laplacian in the frequency domain

Inverse DFT of Laplacian in the frequency domain





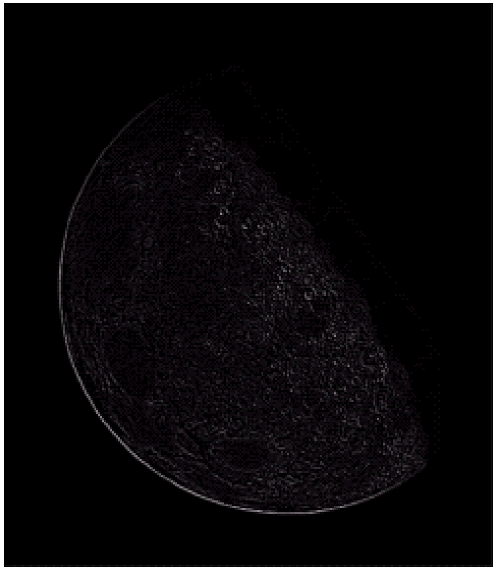


0	1	0
1	−4	1
0	1	0

Zoomed section of the image on the left compared to spatial filter

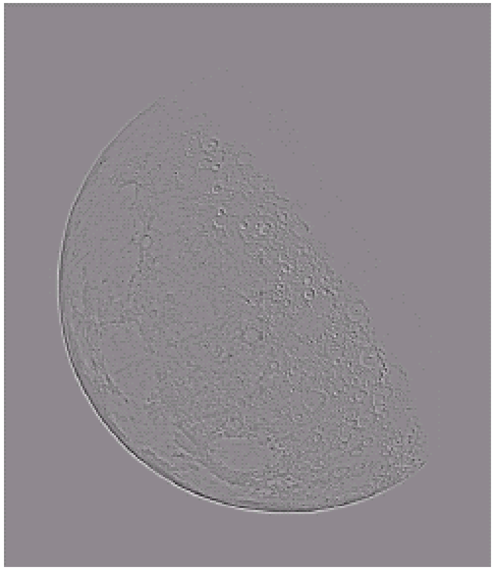
Frequency Domain Laplacian Example

Original image



Laplacian filtered image

Laplacian image scaled



Enhanced image

Fast Fourier Transform

The reason that Fourier based techniques have become so popular is the development of the *Fast Fourier Transform (FFT)* algorithm

Allows the Fourier transform to be carried out in a reasonable amount of time

Reduces the amount of time required to perform a Fourier transform by a factor of 100 – 600 times!

Summary

In this lecture we examined image enhancement in the frequency domain

- The Fourier series & the Fourier transform
- Image Processing in the frequency domain
  - Image smoothing(LPF).
  - Image sharpening(HPF).
- Fast Fourier Transform

Next time we will begin to examine image restoration using the spatial and frequency based techniques we have been looking at

Digital Image Processing

Image Segmentation:
Points, Lines & Edges

Contents

So far we have been considering image processing techniques used to transform images for human interpretation

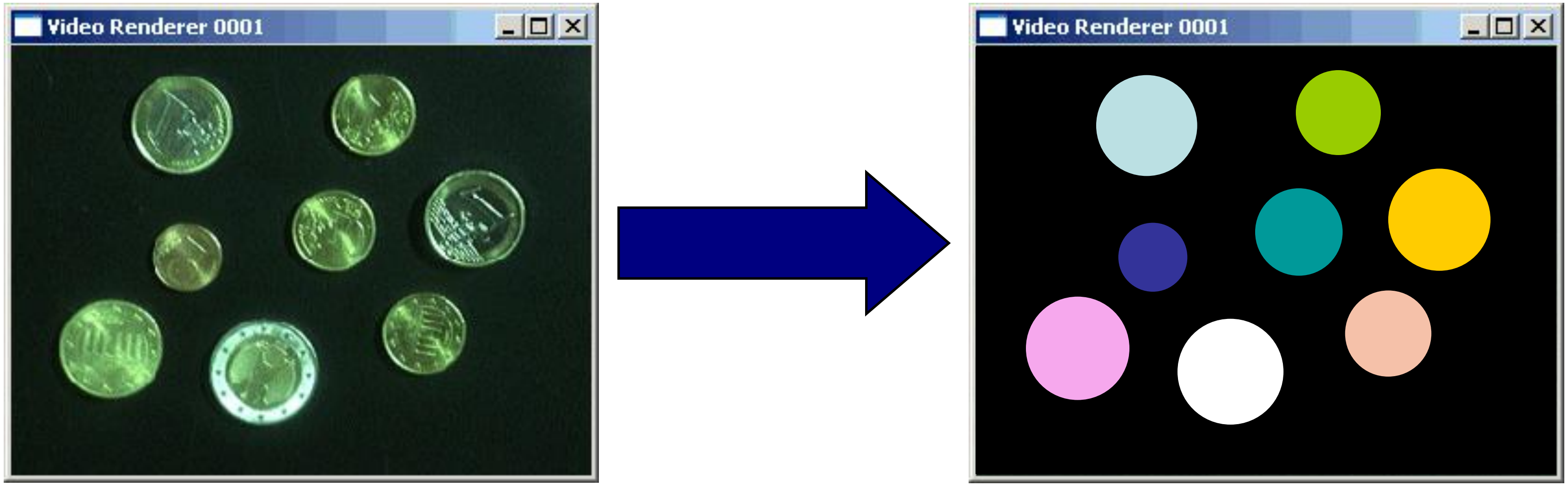
Today we will begin looking at automated image analysis by examining the thorny issue of image segmentation:

- – The segmentation problem
- – Finding points, lines and edges

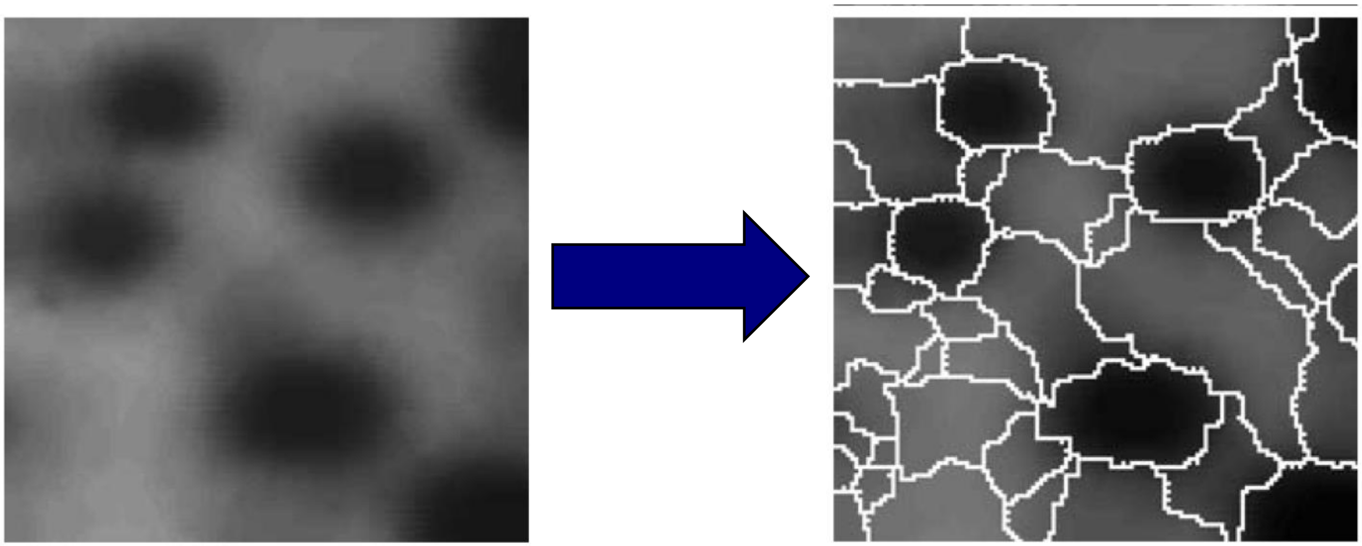
The Segmentation Problem

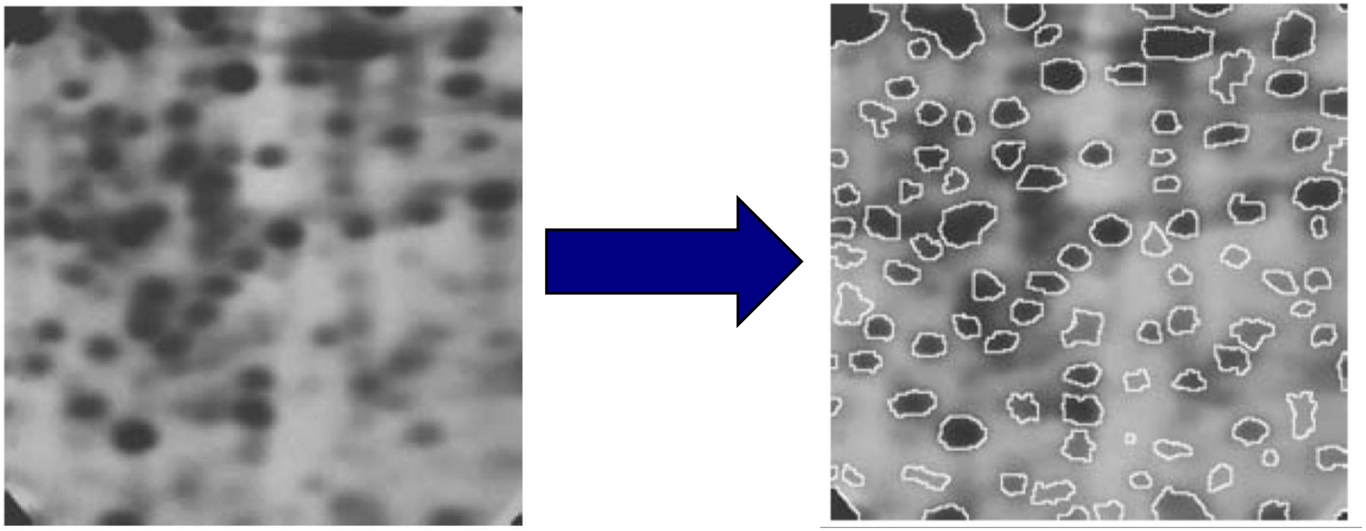
Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image

Typically the first step in any automated computer vision application



Segmentation Examples





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Detection Of Discontinuities

There are three basic types of grey level discontinuities that we tend to look for in digital images:

- Points
- Lines
- Edges

We typically find discontinuities using masks and correlation

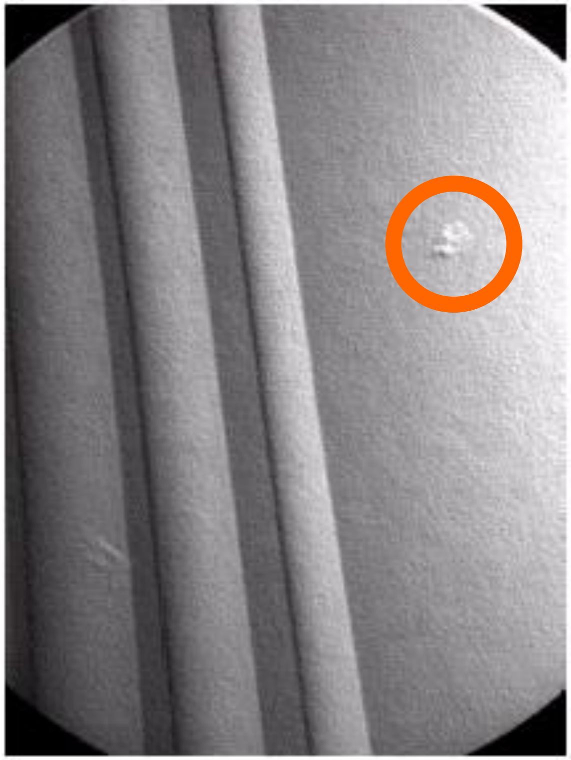
Point Detection

Point detection can be achieved simply using the mask below:

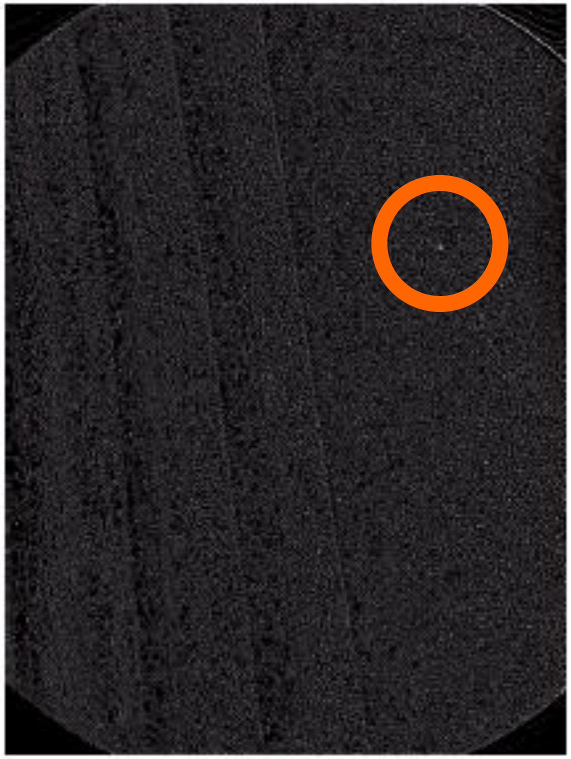
–1	–1	–1
–1	8	–1
–1	–1	–1

Points are detected at those pixels in the subsequent filtered image that are above a set threshold

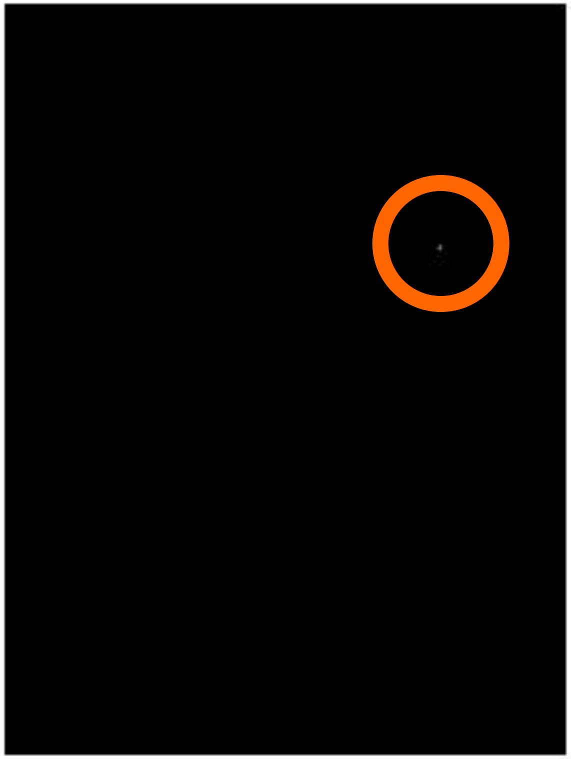
Point Detection (cont…)



X-ray image of a turbine blade



Result of point detection



Result of thresholding

Line Detection

The next level of complexity is to try to detect lines

The masks below will extract lines that are one pixel thick and running in a particular direction

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

−1	2	−1
−1	2	−1
−1	2	−1

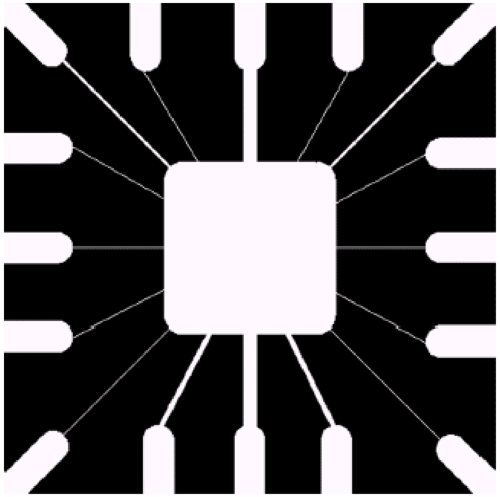
Vertical

2	−1	−1
−1	2	−1
−1	−1	2

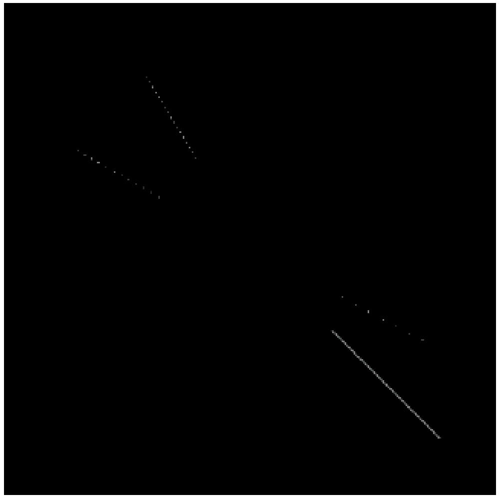
−45°

Line Detection (cont…)

Binary image of a wire bond mask



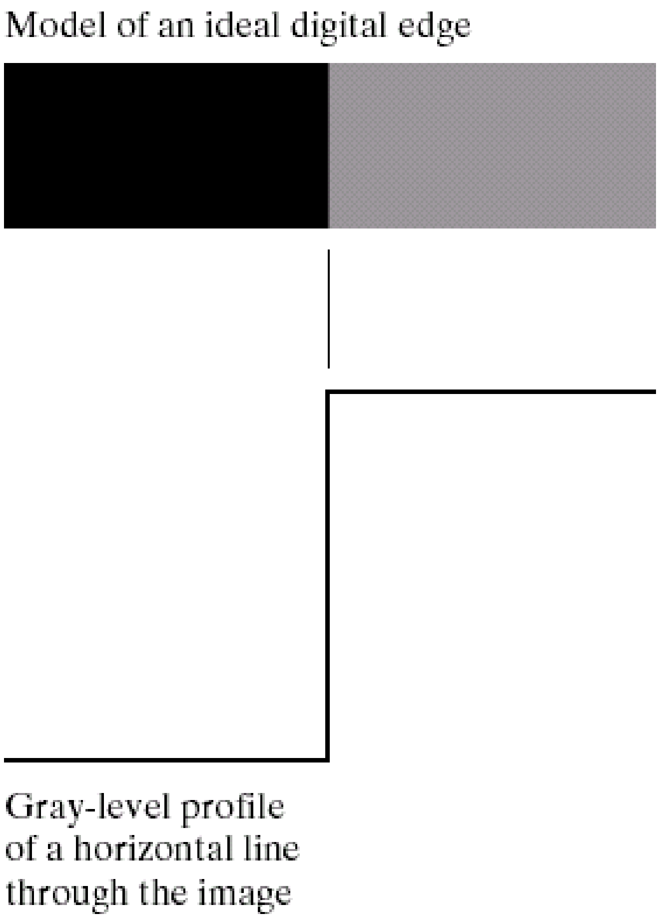
After processing with -45° line detector

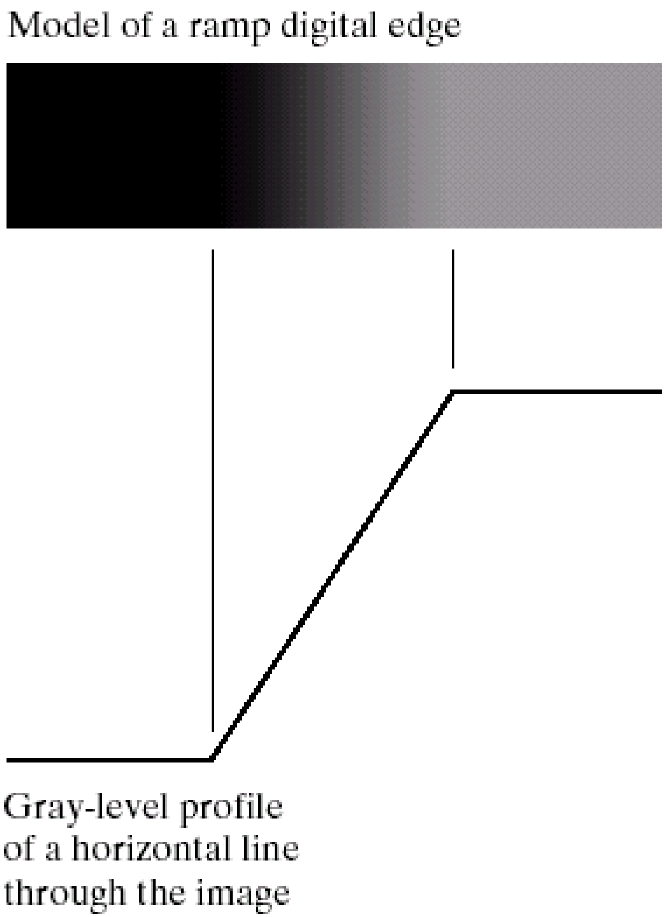


Result of thresholding filtering result

Edge Detection

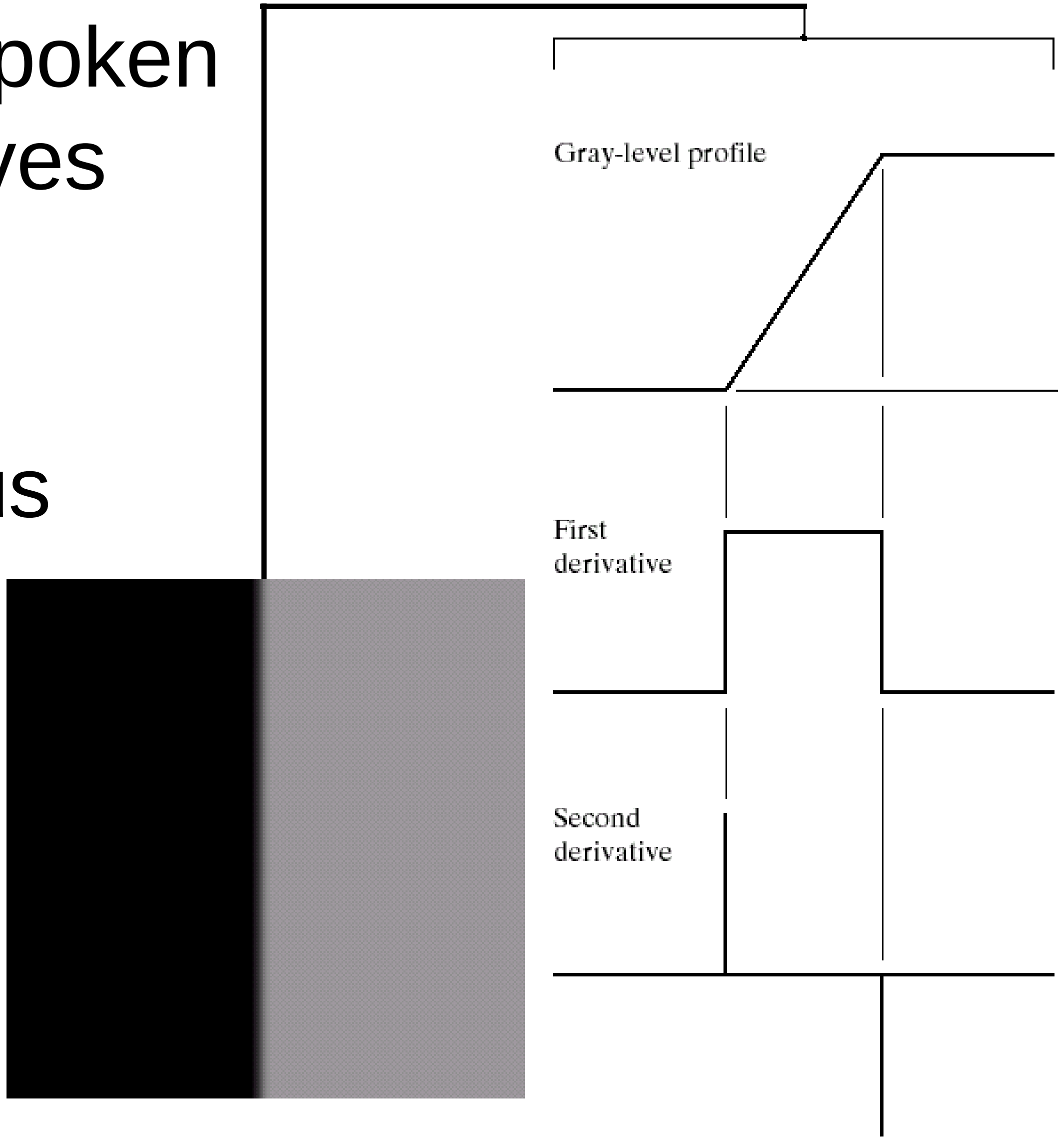
An edge is a set of connected pixels that lie on the boundary between two regions





Edges & Derivatives

- We have already spoken about how derivatives are used to find discontinuities
- 1st derivative tells us where an edge is
- 2nd derivative can be used to show edge direction



Common Edge Detectors

Given a 3*3 region of an image the following edge detection filters can be used

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	−1	−1	−1	0	1
0	0	0	−1	0	1
1	1	1	−1	0	1

Prewitt

−1	0	0	−1
0	1	1	0

Roberts

−1	−2	−1	−1	0	1
0	0	0	−2	0	2
1	2	1	−1	0	1

Sobel

Edge Detection Example

Original Image

Horizontal Gradient Component

Vertical Gradient Component

Combined Edge Image

Edge Detection Problems

Often, problems arise in edge detection in that there are is too much detail

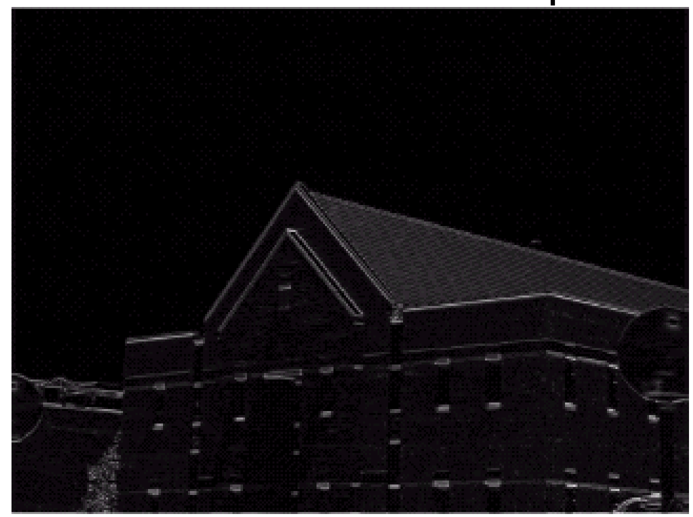
For example, the brickwork in the previous example

One way to overcome this is to smooth images prior to edge detection

Edge Detection Example With Smoothing

Original Image

Horizontal Gradient Component



Vertical Gradient Component

Combined Edge Image

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Laplacian Edge Detection

We encountered the 2nd-order derivative based Laplacian filter already

0	−1	0
−1	4	−1
0	−1	0

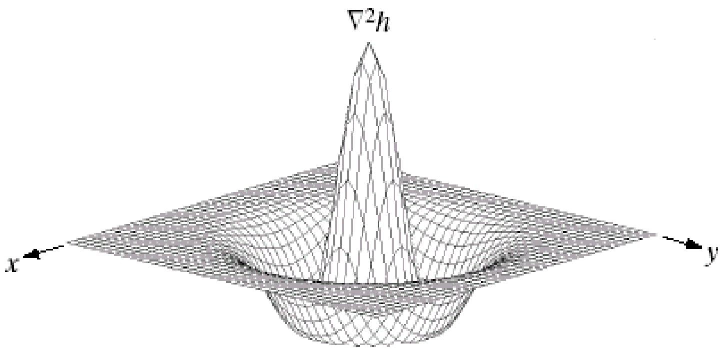
−1	−1	−1
−1	8	−1
−1	−1	−1

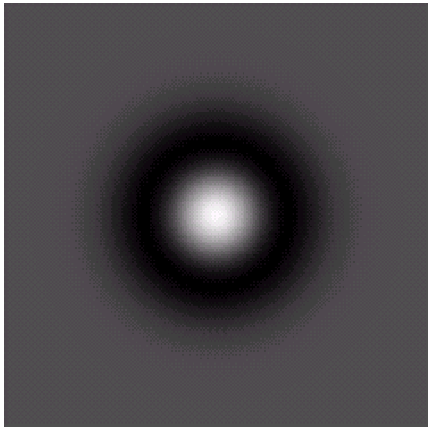
The Laplacian is typically not used by itself as it is too sensitive to noise

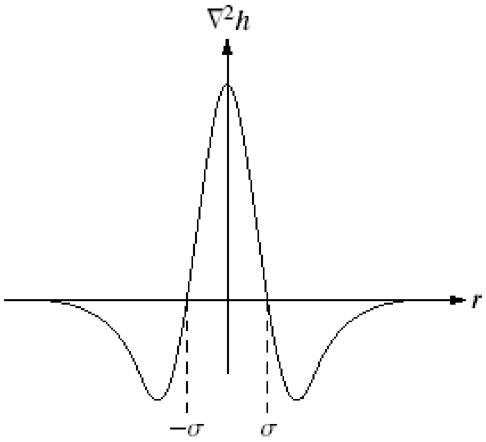
Usually hen used for edge detection the Laplacian is combined with a smoothing Gaussian filter

Laplacian Of Gaussian

The Laplacian of Gaussian (or Mexican hat) filter uses the Gaussian for noise removal and the Laplacian for edge detection

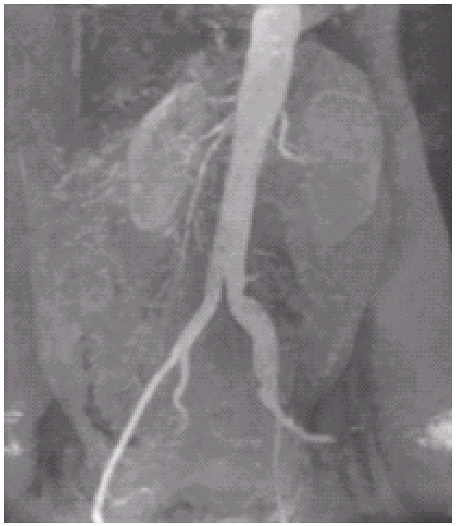


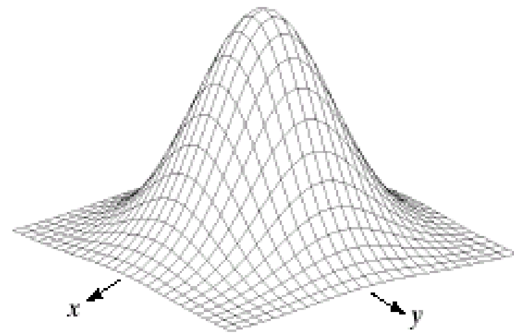




0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Laplacian Of Gaussian Example





−1	−1	−1
−1	8	−1
−1	−1	−1

Summary

In this lecture we have begun looking at segmentation, and in particular edge detection

Edge detection is massively important as it is in many cases the first step to object recognition

Digital Image Processing

Image Segmentation: Thresholding

Contents

Today we will continue to look at the problem of segmentation, this time though in terms of thresholding

In particular we will look at:

- What is thresholding?
- Simple thresholding
- Adaptive thresholding

Thresholding

Thresholding is usually the first step in any segmentation approach

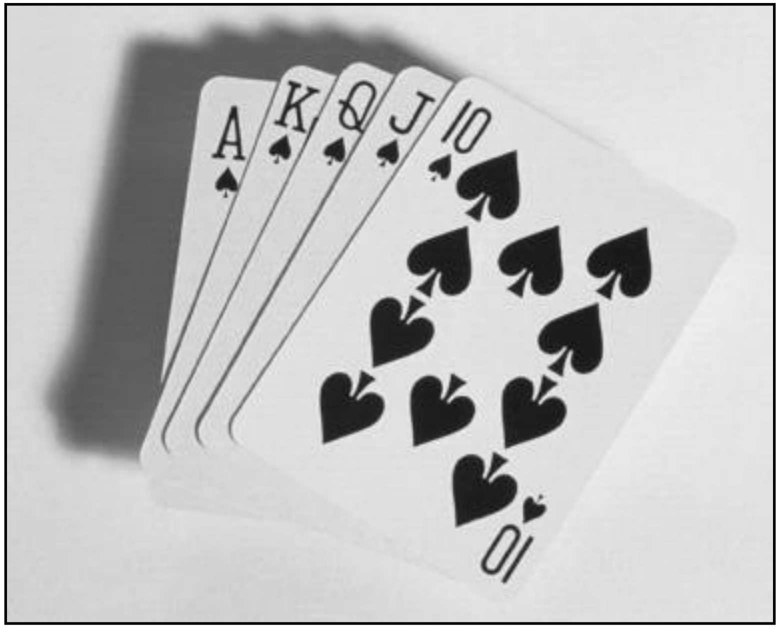
We have talked about simple single value thresholding already

Single value thresholding can be given mathematically as follows:

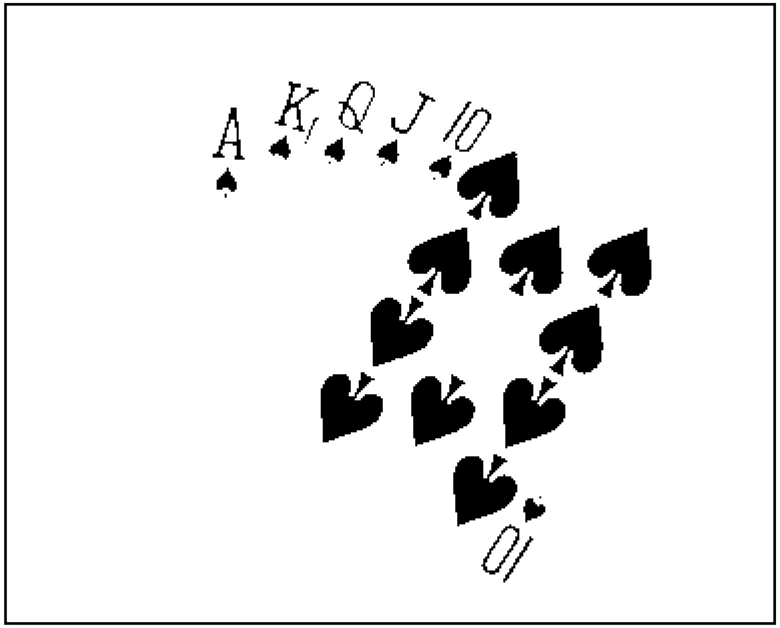
$$g(x,y) = \begin{cases} 1\, if\;\; f(x,y) > T \\ 0\, if\;\; f(x,y) \leq T \end{cases}$$

Thresholding Example

Imagine a poker playing robot that needs to visually interpret the cards in its hand



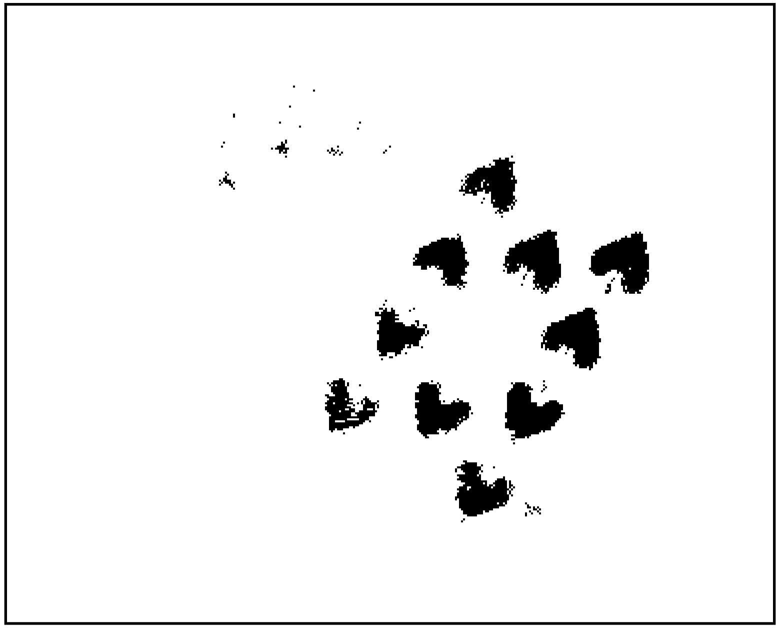
Original Image



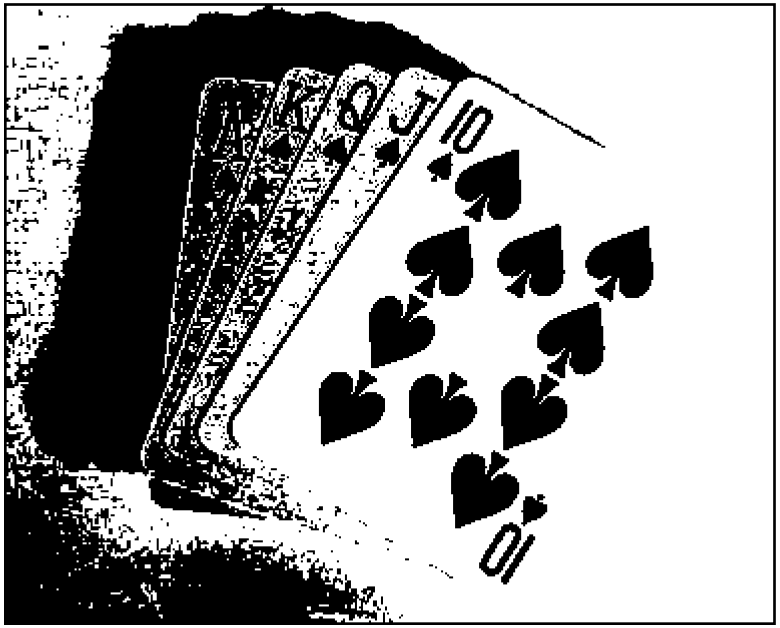
Thresholded Image

But Be Careful

If you get the threshold wrong the results can be disastrous



Threshold Too Low



Threshold Too High

Basic Global Thresholding

- Based on the histogram of an image.
- Partition the image histogram using a single global threshold.

The success of this technique very strongly depends on how well the histogram can be partitioned

Basic Global Thresholding Algorithm

The basic global threshold, T, is calculated as follows:

1. Select an initial estimate for T (typically the average grey level in the image)
2. Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels ≤ T
3. Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2

Basic Global Thresholding Algorithm

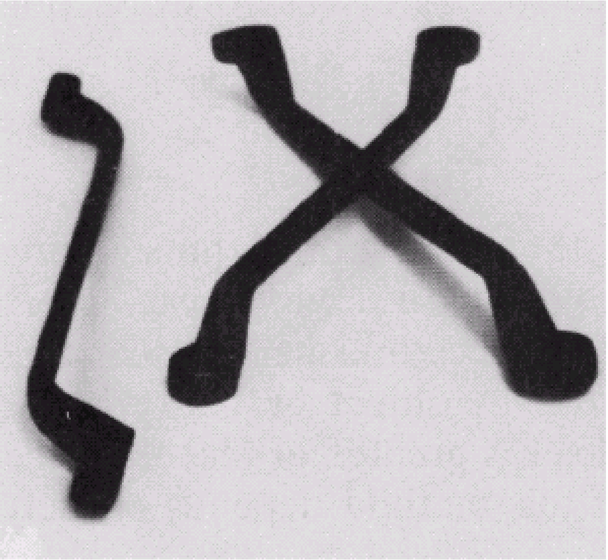
4. Compute a new threshold value:

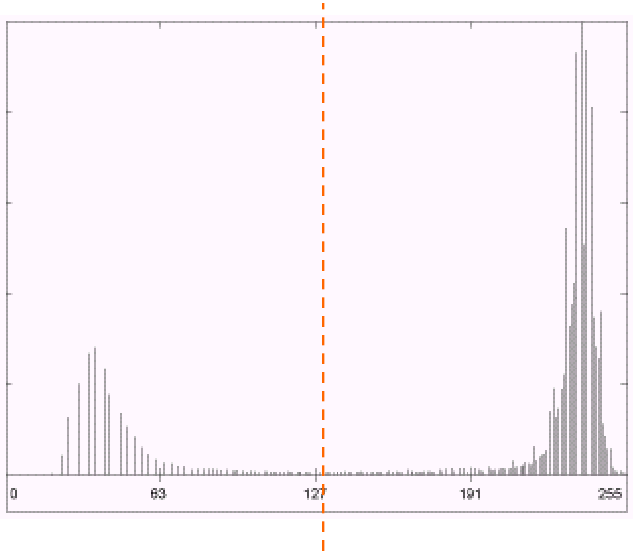
$$T = \frac{\mu_1 + \mu_2}{2}$$

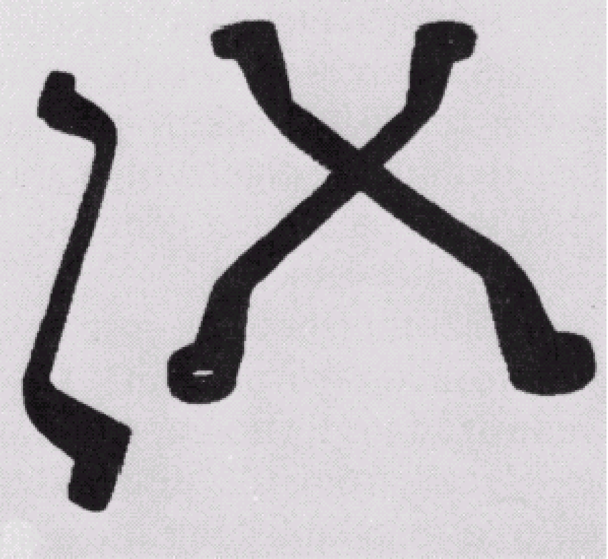
5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit T_∞

This algorithm works very well for finding thresholds when the histogram is suitable

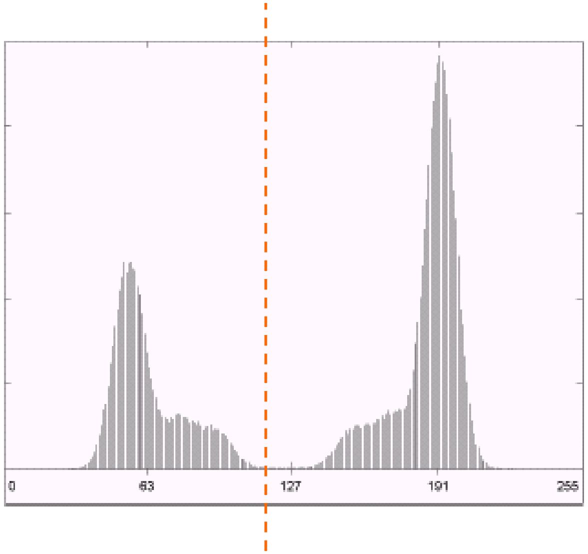
Thresholding Example 1







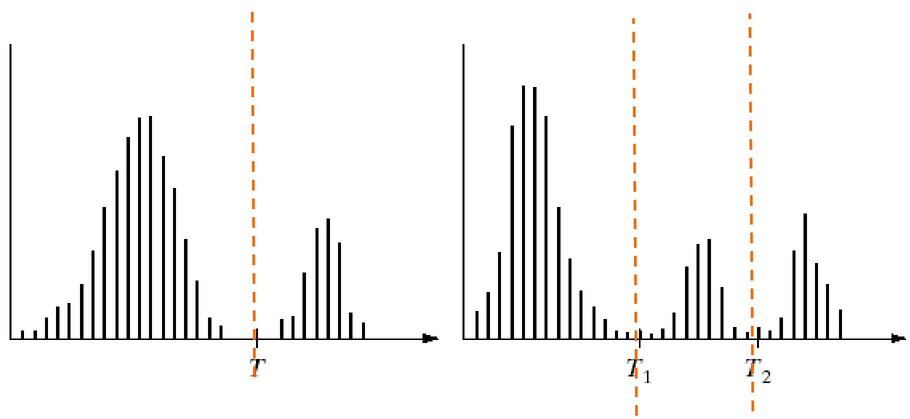
Thresholding Example 2



Problems With Single Value Thresholding

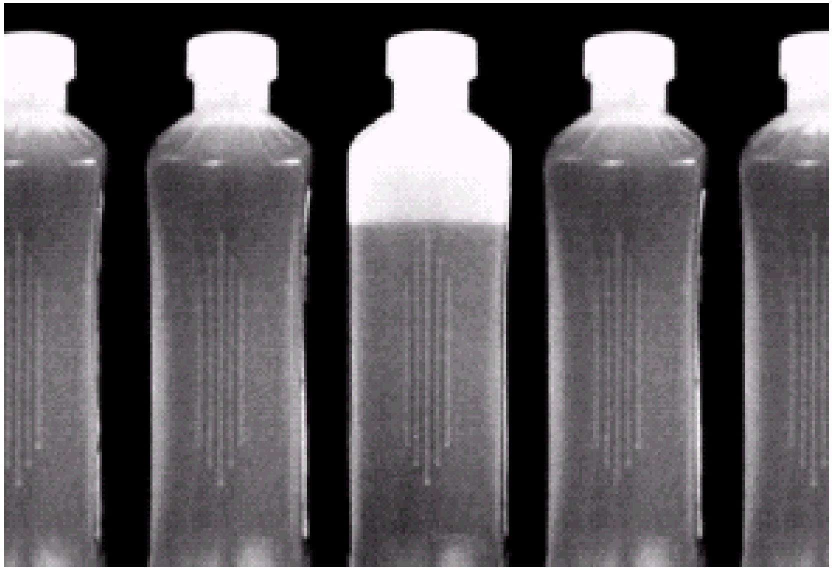
Single value thresholding only works for bimodal (نمطی-شکلی)histograms

Images with other kinds of histograms need more than a single threshold



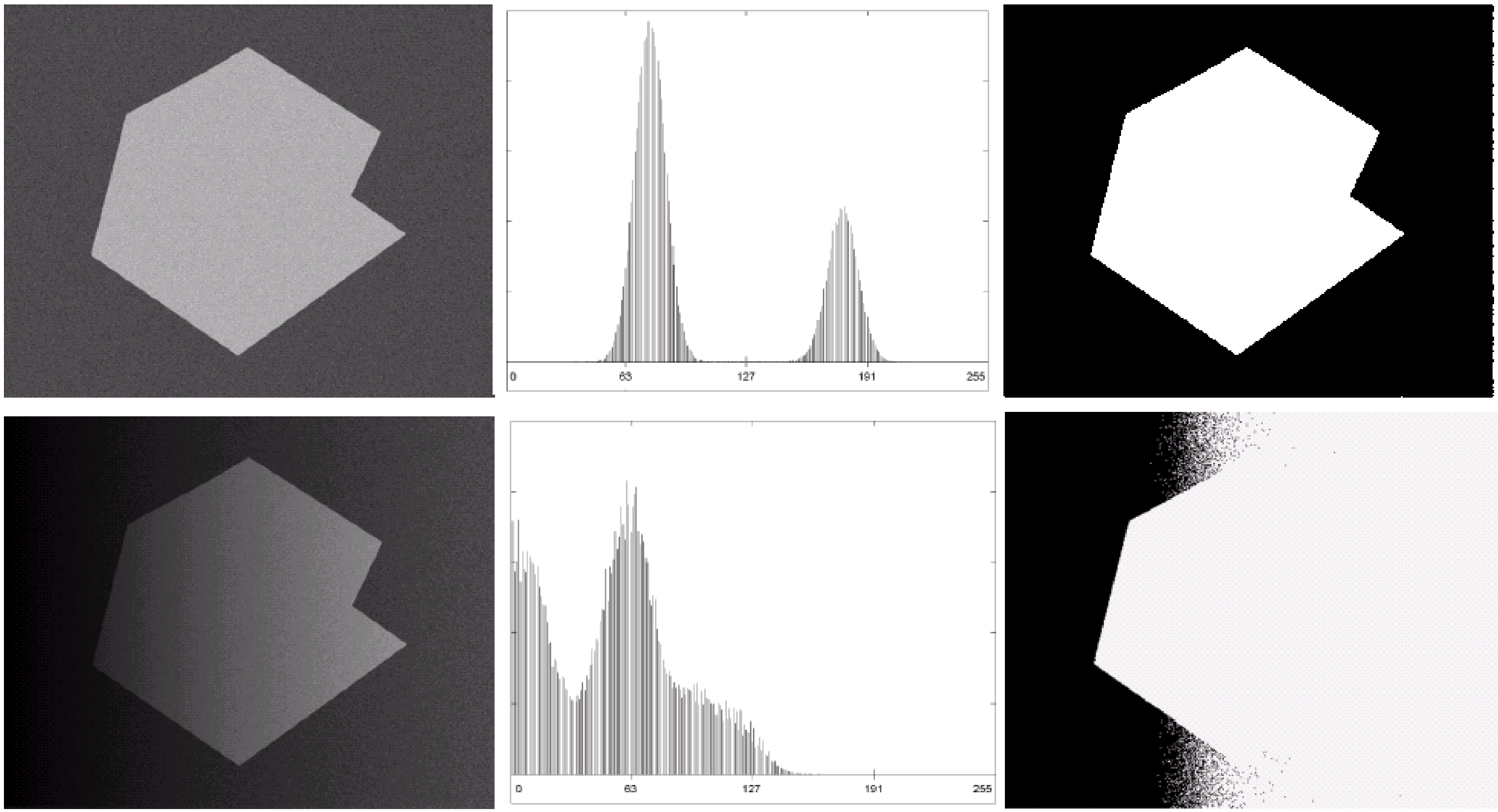
Problems With Single Value Thresholding (cont…)

- Let's say we want to isolate the contents of the bottles
- Think about what the histogram for this image would look like



- What would happen if we used a single threshold value?

Single Value Thresholding and Illumination



Uneven illumination can really upset a single valued thresholding scheme

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

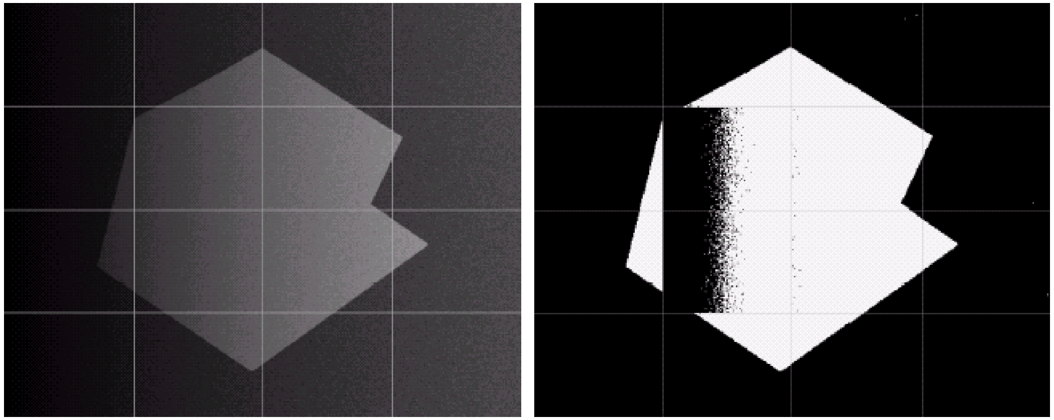
Basic Adaptive Thresholding

An approach to handling situations in which single value thresholding will not work is to **divide an image into sub images and threshold these individually**

Since the threshold for each pixel depends on its location within an image this technique is said to *adaptive*

Basic Adaptive Thresholding Example

The image below shows an example of using adaptive thresholding with the image shown previously



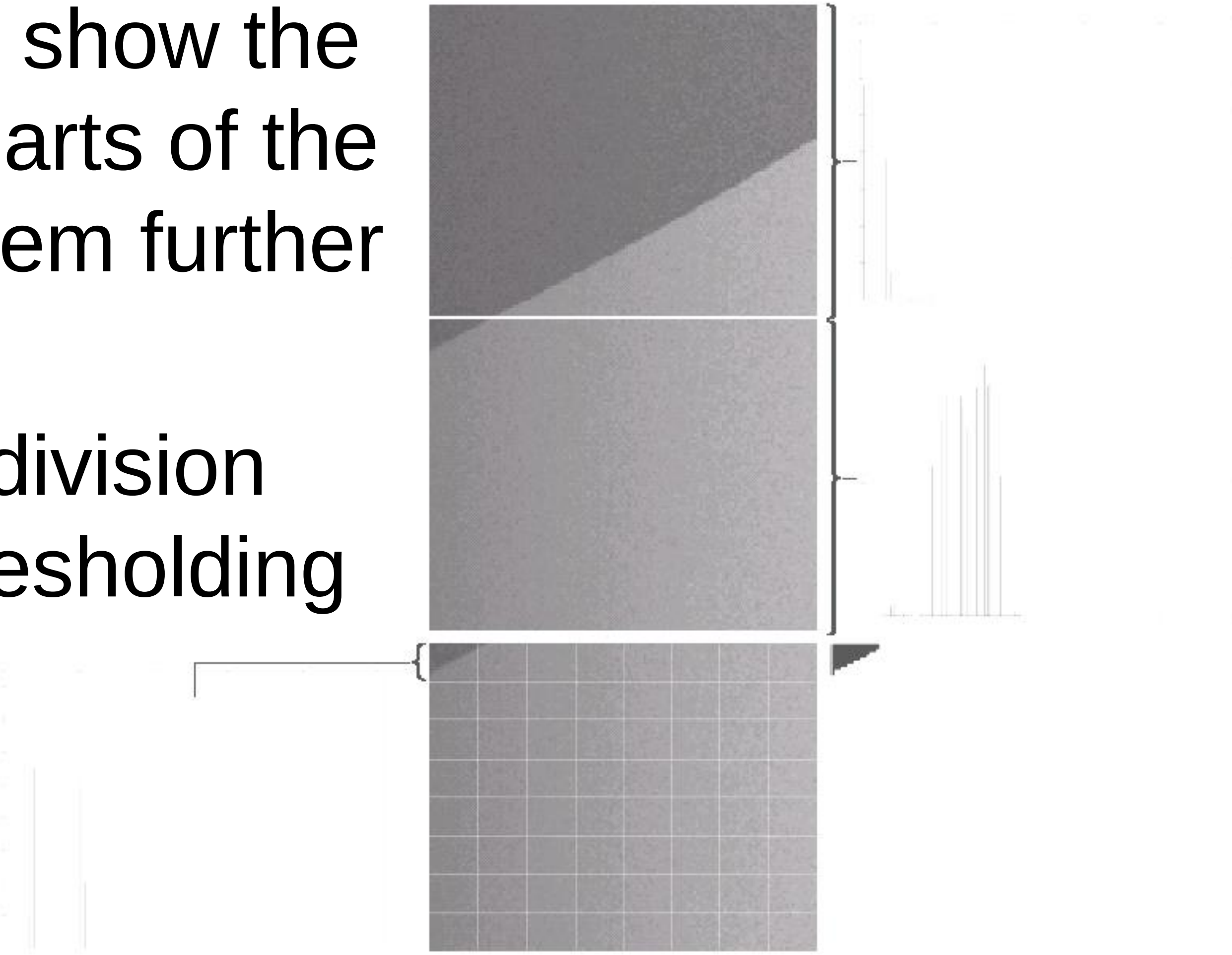
As can be seen success is mixed

But, we can further subdivide the troublesome sub images for more success

Basic Adaptive Thresholding Example (cont…)

These images show the troublesome parts of the previous problem further subdivided

After this sub division successful thresholding can be achieved



Four types of segmentation algorithms

Thresholding - *Similarity*
Based on pixel intensities (shape of histogram is often used for automation).

Edge-based - *Discontinuity*
Detecting edges that separate regions from each other.

Region-based - *Similarity*
Grouping similar pixels (with e.g. region growing or merge & split).

Watershed segmentation - *Discontinuity*
Find regions corresponding to local minima in intensity.

Match-based - *Similarity*
Comparison to a given template.

Summary

In this lecture we have begun looking at segmentation, and in particular thresholding

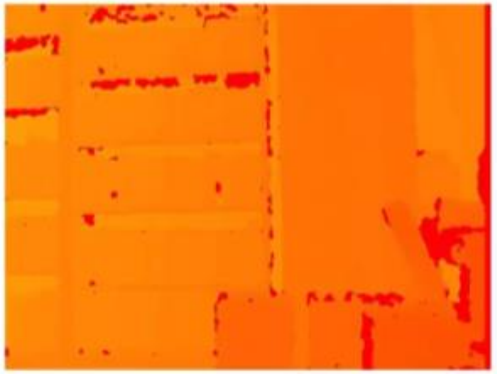
We saw the basic global thresholding algorithm and its shortcomings

We also saw a simple way to overcome some of these limitations using adaptive thresholding

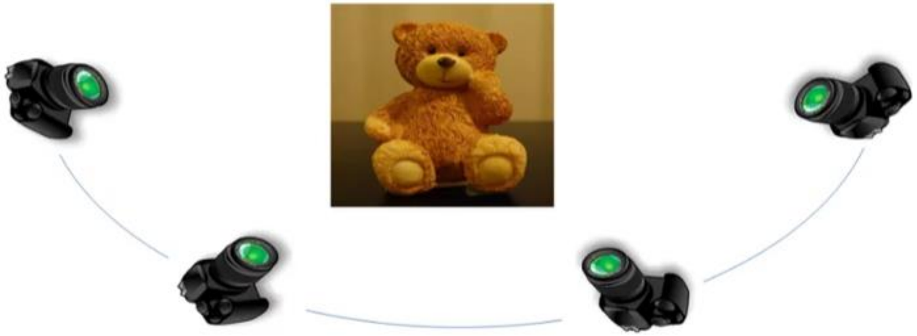
Video Processing

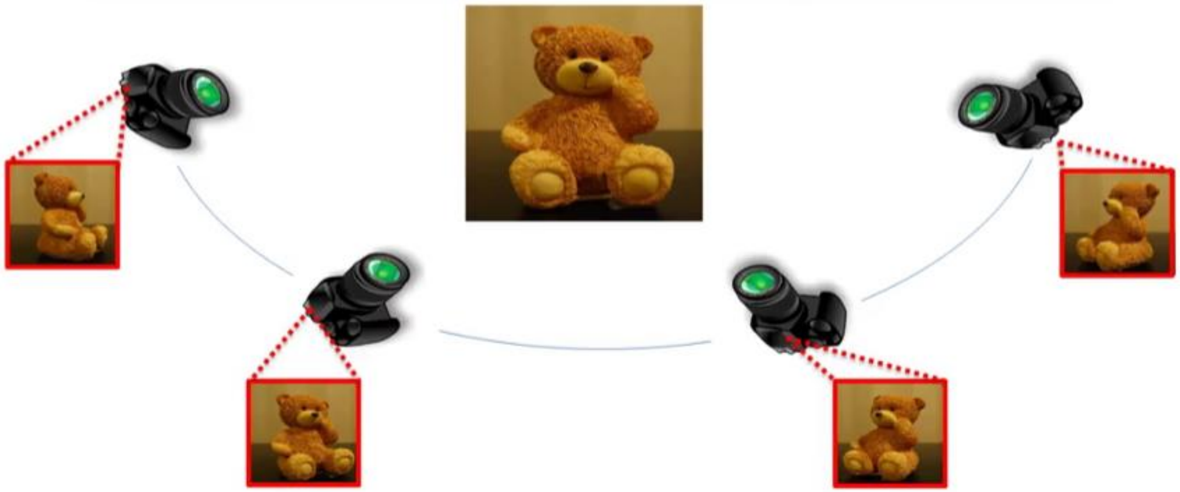
3D: Videos

- Video is a 3D signal. It has two spacial and one tempora coordinates, while a 3D volume has three spatial coordinate x y, z.
- Video can show both the visible image as well as the depth image acquired by the kinect camera.
- Very often we're interested in capturing the three-dimensiona structure of an object. So, instead of using two cameras as ir the stereo case, we use many cameras on a specific rig. So as you can see here, the image of this particular object is viewed from many different angles.
- As an example of a four dimensional signal >> is looking a the objects volume.



Multi-Camera Imaging



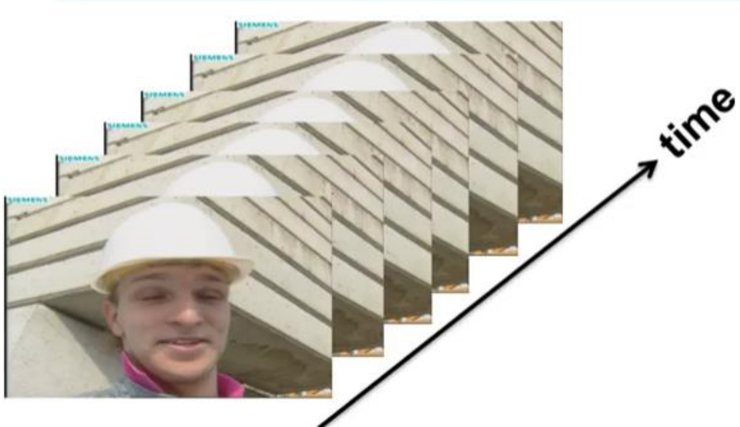


4D Representation

- Typically, x, y, z signal that changes over time, so time is the fourth independent variable.
- Some of the tools that we use to describe signals carry over from 1D to 2D to MD is the straight forward extension one just adds one more variable, and everything remains the same in some sense.

Videos

- A video consists of individual frames, and one could argue that a video is nothing else than a collection of images.
- Therefore, if we have an outgoing that is effective in processing an image, a still frame, as it's called. Also, we could apply the same algorithm to frame after frame.
- However, what is special about video is that these frames are highly correlated and therefore, we can gain if in processing such frames. Thus, this correlation should be taken into account. Of course these frames, if they're displayed at some frame rate, 30 frames per second, for example, one can perceive the actual motion indices.



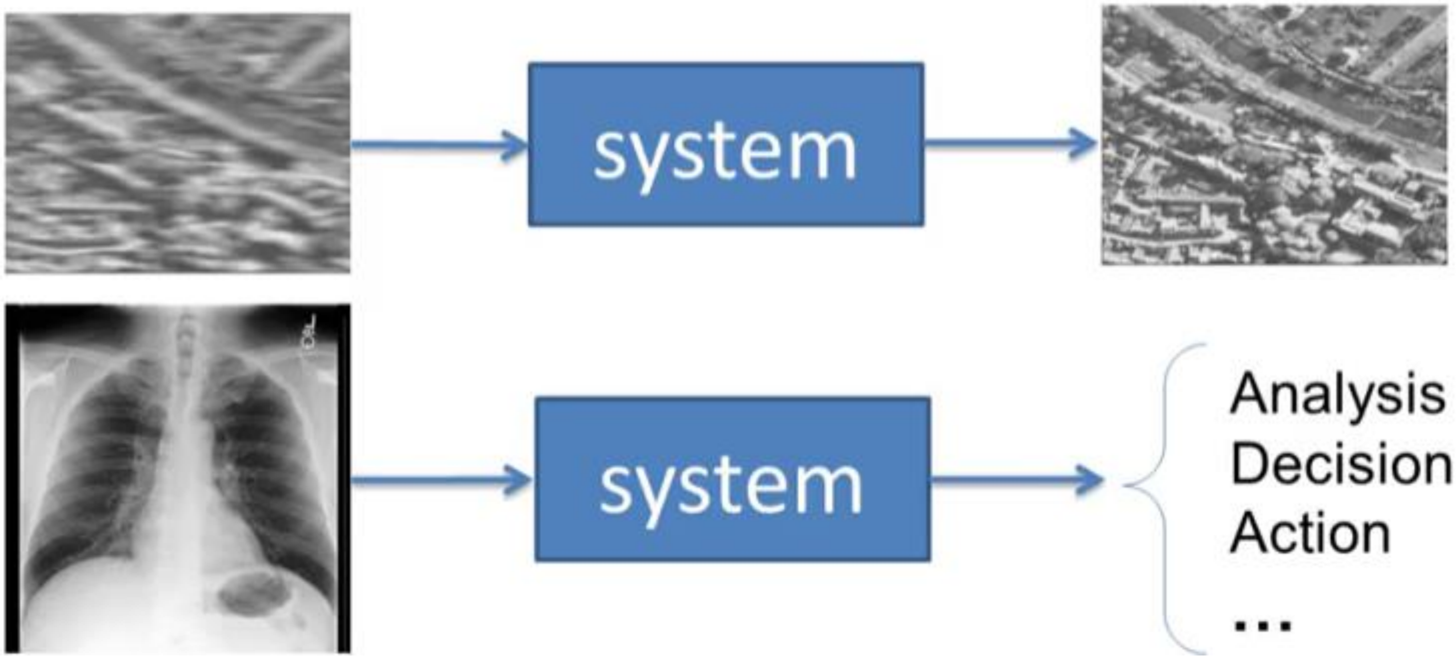
Processing system

- The narrow definition of a processing system is one that accepts an image at the input and generates another image at the output.
- The broader definition however, the one we adopt in this course, is that of a system which for an image at the input might generate an image at the output or a decision based on the analysis of the image, which might result in an action such as to operate for example in a patient if there is a medical image.
- Or the extraction or segmentation of an object from an image based on its color or it's motion and maybe even its classification.

Processing System

- Images and videos are clearly the focus of this class and the images are two dimensional signals but can also be three dimensional are in the multispectral, hyperspectral images while video is a three dimensional signal. So we are a dimension.
- Finally, processing means the manipulation of the values of an image or a video by computer. So as the resulting image is more useful to us, or has some desirable properties.
- For example the result of processing might be the removal of blur, as is the case here you, you see, the input to the system is an aerial photograph that is blurred with the motion between the camera and the scene and the output is a sharpened, a restored image.

Image and Video Processing



- Now if an image is input, an image is output this has been the kind of narrow definition of processing, or so but if we do as filtering, and the broader meaning of processing that we adopt here is that an image or video can be input to a system and we're interested in **extracting important features** from such an image or we're interested in making decisions based on the image.
- The best example here shows this is a chest x-ray and the input and based on the analysis that would be performed whether there's a malignant tumor or not, for example, certain decisions and actions will be the output of the system.

Image and Video Processing

- As time goes on, what we see is that the boundary is between tradition and separate areas become **fuzzy** or in other words there's overlap between these traditionally separate areas. So, when it comes to signal processing and more specifically between with the video processing that we'll be dealing with in this class. There is overlap between the, this field and the fields of communication, computer vision, machine learning and optimization.

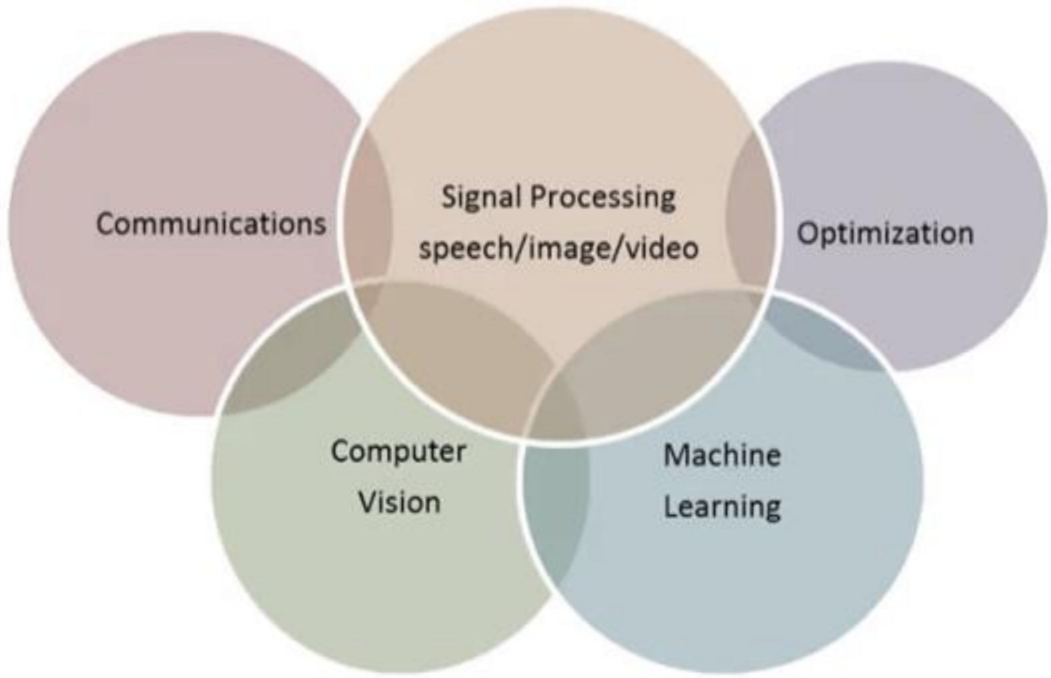


Image Processing

- Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers.
- The goal of this manipulation can be divided into three categories:

❑ Image Processing *image in* → *image out*

❑ Image Analysis *image in* → *measurements out*

❑ Image Understanding *image in* → *high-level description out*

Image Processing

- An image defined in the “real world” is considered to be a function of two real variables, for example, $a(x,y)$ with a as the amplitude (e.g. brightness) of the image at the *real* coordinate position (x,y).
- An image may be considered to contain sub-images sometimes referred to as *regions–of–interest*, *ROIs*, or simply *regions*.
- This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region.
- In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve color rendition.

Image Processing

- The amplitudes of a given image will almost always be either real numbers or integer numbers. The latter is usually a result of a quantization process that converts a continuous range (say, between 0 and 100%) to a discrete number of levels.
- In certain image-forming processes, however, the signal may involve photon counting which implies that the amplitude would be inherently quantized.
- In other image forming procedures, such as magnetic resonance imaging, the direct physical measurement yields a complex number in the form of a real magnitude and a real phase.

Digital Image Definitions

- A digital image $a[m,n]$ described in a 2D discrete space is derived from an analog image $a(x,y)$ in a 2D continuous space through a *sampling* process that is frequently referred to as digitization.

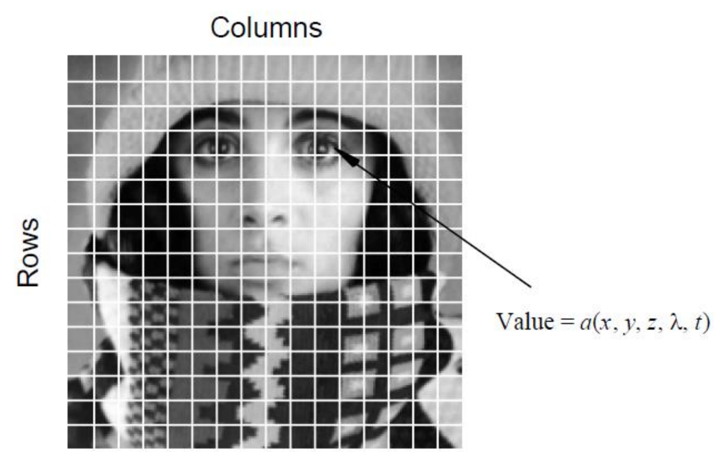
- The 2D continuous image $a(x,y)$ is divided into N *rows* and M *columns*. The intersection of a row and a column is termed a *pixel*.

- The value assigned to the integer coordinates $[m,n]$ with $\{m=0,1,2,\ldots,M-1\}$ and $\{n=0,1,2,\ldots,N-1\}$ is $a[m,n]$.

- In fact, in most cases $a(x,y)$ – which we might consider to be the physical signal that impinges on the face of a 2D sensor – is actually a function of many variables including depth (z), color (λ), and time (t).

Digital Image Definitions

- The image shown in the Figure has been divided into N = 16 rows and M = 16 columns.
- The value assigned to every pixel is the average brightness in the pixel rounded to the nearest integer value.
- The process of representing the amplitude of the 2D signal at a given coordinate as an integer value with L different gray levels is usually referred to as amplitude quantization or simply *quantization*.



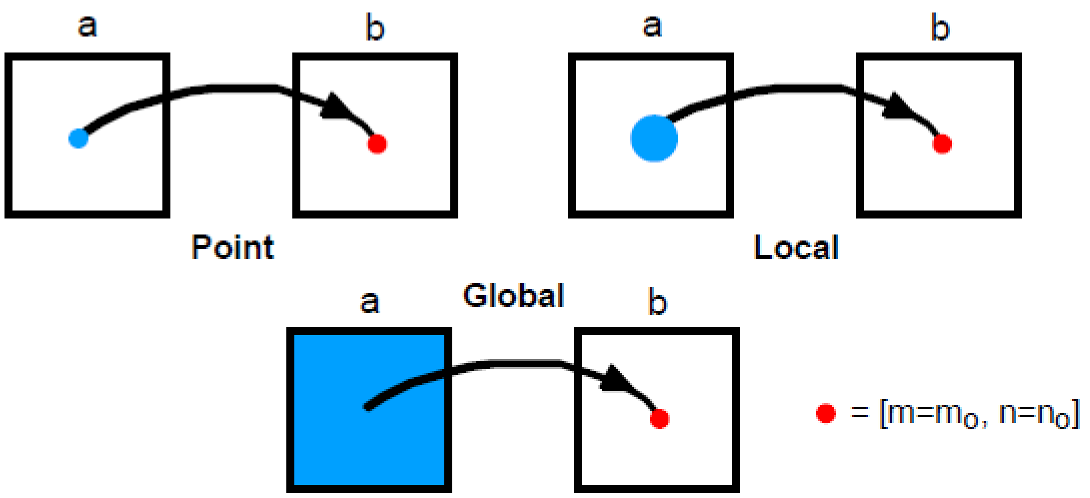
Common Values

Parameter	*Symbol*	*Typical values*
Rows	N	256,512,525,625,1024,1080
Columns	M	256,512,768,1024,1920
Gray Levels	L	2,64,256,1024,4096,16384

- There are standard values for the various parameters encountered in digital image processing. These values can be caused by video standards to keep digital circuitry simple, where M=N=2^K where {K = 8,9,10,11,12}.

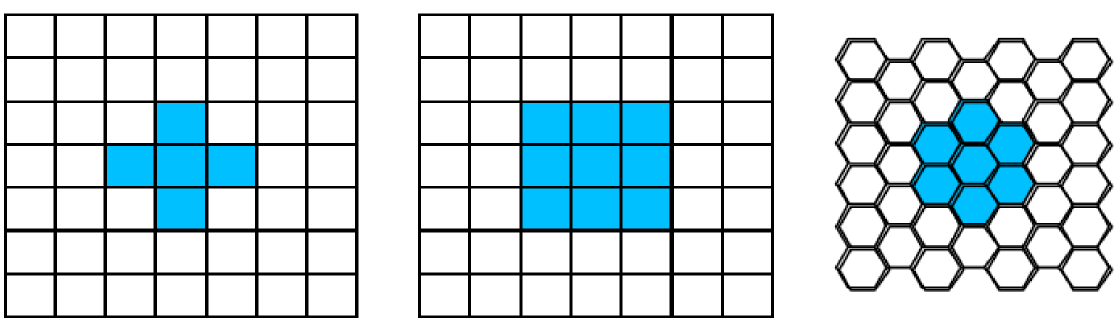
Characteristics of Image Operations

Operation	Characterization	Generic Complexity/Pixel
• *Point*	– the output value at a specific coordinate is dependent only on the input value at that same coordinate.	*constant*
• *Local*	– the output value at a specific coordinate is dependent on the input values in the *neighborhood* of that same coordinate.	P^2
• *Global*	– the output value at a specific coordinate is dependent on all the values in the input image.	N^2



Types of neighborhoods

- Neighborhood operations play a key role in modern digital image processing. It is therefore important to understand how images can be sampled and how that relates to the various neighborhoods that can be used to process an image.
  - **Rectangular sampling** – In most cases, images are sampled by laying a rectangular grid over an image.
  - **Hexagonal sampling**
- Local operations produce an output pixel value *b*[*m*=*mo*,*n*=*no*] based upon the pixel values in the *neighborhood* of *a*[*m*=*mo*,*n*=*no*].
- Some of the most common neighborhoods are the 4-connected neighborhood and the 8-connected neighborhood in the case of rectangular sampling and the 6-connected neighborhood in the case of hexagonal sampling .



Rectangular sampling 4-connected

Rectangular sampling 8-connected

Hexagonal sampling 6-connected

Video Parameters

Standard / *Property*	**NTSC**	**PAL**	**SECAM**
images / second	29.97	25	25
ms / image	33.37	40.0	40.0
lines / image	525	625	625
(horiz./vert.) = aspect ratio	4:3	4:3	4:3
interlace	2:1	2:1	2:1
µs / line	63.56	64.00	64.00

The End